# Block 2

<table>
<thead>
<tr>
<th>STUDY GUIDE-2</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIT 6</td>
<td>5</td>
</tr>
<tr>
<td>Matchings and Covers</td>
<td></td>
</tr>
<tr>
<td>UNIT 7</td>
<td>29</td>
</tr>
<tr>
<td>Connectivity and Networks</td>
<td></td>
</tr>
<tr>
<td>UNIT 8</td>
<td>53</td>
</tr>
<tr>
<td>Colouring of Graphs</td>
<td></td>
</tr>
<tr>
<td>UNIT 9</td>
<td>65</td>
</tr>
<tr>
<td>Planar Graphs</td>
<td></td>
</tr>
<tr>
<td>UNIT 10</td>
<td>79</td>
</tr>
<tr>
<td>Hamiltonian Graphs</td>
<td></td>
</tr>
<tr>
<td>Appendix</td>
<td></td>
</tr>
<tr>
<td>List of Practical Sessions</td>
<td>93</td>
</tr>
</tbody>
</table>
Curriculum Design Committee

Dr. B.D. Acharya
Dept. of Science & Technology, New Delhi
Prof. Adimurthi
School of Mathematics
TIFR, Bangalore
Prof. Archana Aggarwal
CESP, School of Social Sciences
JNU, New Delhi
Prof. R.B. Bapat
Indian Statistical Institute
New Delhi
Prof. M.C. Bhandari
Dept. of Mathematics
IIT, Kanpur
Prof. R. Bhatia
Indian Statistical Institute, New Delhi
Prof. A.D. Dharmadhikari
Dept. of Statistics
University of Pune
Prof. O.P. Gupta
Dept. of Financial Studies
University of Delhi
Prof. S.D. Joshi
Dept of Electrical Engineering
IIT, Delhi

Dr. R.K. Khanna
Scientific Analysis Group, DRDO, Delhi
Prof. Susheel Kumar
Dept. of Management Studies
IIT, Delhi
Prof. Veni Madhavan
Scientific Analysis Group
DRDO, Delhi
Prof. J.C. Misra
Dept. of Mathematics
IIT, Kharagpur
Prof. C. Musili
Dept. of Mathematics and Statistics
University of Hyderabad
Prof. Sankar Pal
ISI, Kolkata
Prof. A.P. Singh
PG Dept. of Mathematics
University of Jammu

Course Design Committee

Prof. C. A. Murthy
ISI, Kolkata
Prof. S. B. Pal
IIT, Kharagpur
Prof. C. E. Veni Madhavan
IISc, Bangalore

Dr. B. S. Panda
IIT, Delhi
Prof. Parvin Sinclair
Prof. Poornima Mital
Dr. Atul Razdan
Dr. Deepika
Dr. Atul Razdan

Block Preparation Team

Prof. Sane (Editor)
Dept. of Mathematics
University of Bombay, Mumbai
Dr.Vijayalakshmi
Dept of Mathematics, Anna University
MTT Campus, Chennai

Dr. Parvathi
Dept. of Mathematics
Vimala College, Calicut University, Kerala
Dr. B. Radha Krishnan (Consultant)
Lecturer (Selection Grade)
Dept. of Mathematics
D.B. College, Sasthamcott, Kollan District
Kerala

Prof. Poornima Mital
Dr. S. Venkataraman
Dr. Deepika
Dr. Atul Razdan

Faculty Members

School of Sciences, IGNOU
Prof. Poornima Mital
Dr. Atul Razdan
Prof. Parvin Sinclair
Prof. Sujatha Varma
Dr. S. Venkataraman

Course Coordinator: Prof. Sujatha Varma

Acknowledgements To: Dr. Deepika for her comments on the manuscript.
Mr. Bhagat Singh for typing the manuscript.

© Indira Gandhi National Open University, 2011
ISBN-978-81-266
All right reserved. No part of this work may be reproduced in any form, by mimeograph or any other means, without permission in writing from the Indira Gandhi National Open University.
Further information on the Indira Gandhi National Open University courses, may be obtained from the University’s Office at Maidan Garhi, New Delhi-110 068.
Printed and published on behalf of the Indira Gandhi National Open University, New Delhi by the Director, School of Sciences.
In the previous Block, we familiarised you with some fundamental concepts in Graph Theory such as paths, cycles, trees and degree of a vertex. We also discussed planar graphs, connected graphs, weighted graphs and certain algorithms also.

In this block, we shall familiarise you with some more of concepts. We shall start with matching and covers in Unit 6. A matching in a Graph G is a set of edges of G, no two of which have a vertex in common. Here, we shall illustrate the concepts of perfect matching, maximum matching, maximal matching, alternating paths and augmenting paths. We also discuss certain important theorems due to Berge and, a theorem due to Hall and another theorem called Min-Max theorem. Certain other parameters such as independent sets, vertex cover and edge are also discussed here.

In Unit 7, we discuss connectivity of a graph which is a concept having much applications in network theory. For any graph G, two graph invariants namely vertex connectivity and edge connectivity are helpful in network analysis. An important theorem due to Menger, concepts of blocks, k-connected graphs and k-edge connected graphs are also discussed.

Unit 8 deals with colouring of graphs. Colouring of graphs is a very important topics in Graph Theory and involves colouring of vertices and colouring of edges and faces. We shall discuss colouring of faces in the next unit. We have omitted colouring of edges as it is not included in the syllabus. Here we discuss colouring of vertices. Colouring of vertices is an assignment of colours to the vertices of a graph such that no pair of adjacent vertices are assigned same colours. The related notions such as chromatic number and chromatic graphs are also considered in this unit.

In Unit 9, we discuss planar graphs. The study of the concept of planarity was motivated by the famous Four Colour problem. A brief history of the problem is given as an Appendix with this unit. Here we will investigate the characteristics of planar graphs. We shall also discuss the related concepts such as dual graph and colouring of planar graphs and give a proof of the five colour theorem.

In the last unit of this block, we will focus our attention on Hamiltonian graphs. A spanning cycle in a graph is called a Hamiltonian cycle and a graph having a Hamiltonian cycle is called a Hamiltonian graph. There is no general criteria which characterizes Hamiltonian graphs, as reported so far. But there are conditions which are sufficient for a graph to be Hamiltonian and there are conditions which are necessary for a graph to be Hamiltonian. Here we discuss some of them.

As we already mentioned in the course introduction, this course has a practical component. The objective of the practical component is to give you some practical experience of some of the algorithm learnt in this course. There are 8 practical sessions of 3 hours duration each. The details are given in the Appendix given at the end of this block. All these sessions are compulsory. Apart from this we also advise you to attempt other example problems discussed in the Textbook DW, to get more practice.
Matching theory was introduced by Philip Hall in 1935. This theory has application in solving assignment problem and many interesting combinatorial problems.

In this unit, we discuss the material for discussing the problem posed in Example 1.1.9 of Textbook DW. The problem is about possibility of filling up of job vacancies with suitably qualified candidates from among a set of applicants each of whom is qualified for one or more of the jobs. This situation is modeled using graphs. The question ‘Whether it is possible to find a satisfactory assignment?’ is represented. For, the concepts of matching, saturated vertex, perfect matching, alternating path, augmenting path and Hall’s condition are discussed. Related concepts of maximum matching, maximal matching, a min-max theorem useful in network theory, concepts of independent sets, vertex cover and some related graph parameters are also discussed.

In section 6.2, we give the definitions and illustrations of matching and perfect matching in a graph. Section 6.3 deals with maximal matching, maximum matching, alternating paths, augmenting paths and a theorem due to Berge. The main theorem – Hall’s theorem – is discussed in Section 6.4. The subject of section 6.5 is a min-max theorem which gives the relationship between the solutions to a maximization problem and a minimization problem. In Section 6.6 independent set, vertex cover, edge cover, etc. are discussed. This section deals with graph parameters namely maximum size of independent set, maximum size of matching, minimum size of vertex cover and minimum size of edge cover also.

Since loops and multiple edges are insignificant in the study of matchings, in this unit also we confine to simple graphs only unless otherwise stated.

Pre-requisites for this unit are fundamental knowledge of set theory including the concept of set inclusion, and graph theory including characteristics of bipartite graphs.
Objectives

After studying this unit, you shall be able to

- check whether a given graph possesses a perfect matching or not;
- confirm whether a bipartite graph contains a matching that saturates every vertex in one of its partite sets;
- recognise matchings, maximal matchings, maximum matchings and perfect matchings, if any in a given graph;
- check whether a given matching in a graph is a maximal matching/maximum matching/perfect matching;
- check a given matching is maximum or not;
- find vertex covers, edge covers and independent sets;
- estimate the parameters \( \alpha(G), \alpha'(G), \beta(G) \) and \( \beta'(G) \) for a given graph \( G \);
- check for and find a suitable assignment in a given system.

6.2 MATCHING

In this section we shall introduce you to matching and some related concepts.

Let us look at the job assignment problem stated in page 4 of Textbook DW. It is a classical problem that can be solved using graph models.

The problem is stated as follows:

Consider a set A of jobs a company would like to fill and B a set of candidates applied for. The jobs may be of different nature and some of the candidates might have qualified for some of the jobs only, and not for some other. The questions to be considered are whether all the positions can be filled with suitably qualified candidates from among the applicants? And if so, how?

Let us model this problem using graphs:

Consider a graph G with vertex set \( A \cup B \) and an edge joins \( a \in A \) to \( b \in B \) if and only if the candidate corresponding to \( b \in B \) is qualified for the job corresponding to \( a \in A \). So there are no edges joining two vertices in A or two vertices in B.

Now the problem is of finding a set of edges in G containing exactly one edge incident at each vertex of A and no pair of edges have a common end vertex in B. Such a set of edges is a matching.

For example in the following figure (Fig.1) the sets \( M_1 = \{ e_1, e_3, e_5, e_7 \} \) and \( M_2 = \{ e_3, e_4, e_7, e_8 \} \) are suitable choices. But no set of edges containing \( e_2 \) or \( e_8 \) will work.
Definition 1: A matching in a graph G is a subset M of the edge set E(G) such that no pair of edges in M have a common end vertex in G.

Definition 2: A vertex u in a graph G is said to be saturated by a matching M in G [M-saturated] if u is an end vertex of some edge in G.

Definition 3: If M is a matching in a graph G and if all vertices of G are M-saturated then M is called a perfect matching.

Let us consider some examples.

Example 1: Let G be the graph model of the Konigsberg bridge problem shown in page 2 of Textbook DW.

In this graph $M_1=\{e_1, e_7\}$ is a matching and no other subset of edges containing $e_1$ is a matching since every other vertex shares an edge with $e_1$.

Can you find some other matching in this graph? You might see that $M_2=\{e_3, e_6\}$, $M_3=\{e_4\}$ and $M_5=\{e_5\}$ are also matchings in the G.

Here all the four vertices are $M_1$-saturated as well as $M_2$-saturated and hence they are perfect matchings. But the vertices x and y are not $M_3$-saturated and x and z are not $M_5$ saturated. So $M_3$ and $M_5$ are not perfect matchings.

We shall consider another example.

Example 2: Let us consider the Petersen graph defined in 1.1.36, page 13 of Textbook DW with vertices labeled $v_1, v_2, v_3, ..., v_{10}$ and edges labeled $e_1, e_2, e_3, ..., e_{15}$ as in Fig. 3.
M₁={e₆, e₇, e₈, e₉, e₁₀}, M₂={e₁, e₃, e₁₀, e₁₂, e₁₃} and M₃={e₂, e₅, e₉} are matchings. But no three of the five edges on the outer cycle form a matching. Here M₁ and M₂ are perfect matchings as each of them saturates every vertex. But M₃ is not perfect.

You can start reading *Textbook DW*.

**NOTES:**

(i) **Page 107 line 1:**

The problem of pairing room mates: Consider a set of people to be paired to accommodate two each in a room. Here the constraint is not on the number of rooms or number of peoples but the individuals’ willingness to share rooms with others. If everybody is ready to share a room with any other there is no difficulty, if the number of people is even. But that may not be the real situation. Consider the graph model of this problem. The vertex set of the graph corresponds the set of people and edge joins two vertices if and only if the respective persons are ready to share a room. This graph need not be bipartite. Then the question is of finding a matching in a graph.

An obvious necessary condition is that the number of people must be even. The Hall’s theorem can be applied here only if it is possible to find out a spanning bipartite subgraph with partite sets of equal size.

(ii) **Page 107 line 7**

Matching is defined as a set of edges with certain conditions. But you should convince yourself that there is no existence for an edge
in a graph without its end vertices. So, for every mention about edges, the presence of end-vertices is understood. Because of this, a matching in a graph \(G\) can be viewed as a sub-graph of \(G\) whose components are \(K_2\), the complete graph of order two. Refer to the Examples 1 and 2 above.

(iii) **Page 107 line 8**

Sharing end points means the set of end vertices of any two edges are disjoint subsets of the vertex set. A vertex \(v\) is incident to an edge \(e\) means that \(v\) is an end vertex of the edge \(e\).

[refer page 6 lines 11 and 10 from the bottom of Textbook DW]

(iv) **Page 107 lines 9 and 10**

If \(G\) is a graph and \(M\) a matching in it then a vertex \(v\) of \(G\) is \(M\)-saturated if \(M\) contains an edge \(e\) with \(v\) as one of its end vertices. If no such edge is in \(M\) then \(v\) is \(M\)-unsaturated.

(v) **Page 107 from line 12 to line 14**

Recall the definition of \(K_{n,n}\), the complete bipartite graph [page 9 lines 26 and 27 of text book DW]. It has \(n\) vertices in each partite set and every vertex in each partite set is adjacent to all vertices in the other partite set. So it is easy to find a perfect matching in it. For example \(M = \{x_i, y_i / i = 1, 2, 3, ..., n\}\) is a perfect matching.

For finding all perfect matchings in \(K_{n,n}\), consider any permutation \(P\) of the set \(Y\) and take the edge with end vertex \(x_i\) and the \(i^{th}\) vertex in \(P\), for \(i = 1, 2, 3, ..., n\). Among these the perfect matching \(\{x_i, y_i / i = 1, 2, 3, ..., n\}\) correspond the identity permutation. Since the order of arrangement of elements of \(Y\) is different in distinct permutations of it, we get distinct machings. These are the only perfect matchings possible since there are only \(n\) edges incident with each vertex in \(K_{n,n}\) so \(n!\) matchings.

An alternate computation of the number of perfect matchings in \(K_{n,n}\) is given below:

There are \(n\) edges incident at \(x_1\). So we have \(n\) choices to select an edge incident at \(x_1\). Let it be \(x_1y_1\). After selecting it, for an edge incident at \(x_2\) we have exactly \(n-1\) choices, i.e. except \(x_2y_1\).

Similarly, for edges at \(x_3, x_4, \ldots\) etc. we have \(n-2, n-3, \ldots\) choices for successive selection of edges incident at. Hence, by the principle of counting the total choices for perfect matching in \(K_{n,n}\) is \(n!\)

Clearly each matching provides a one-to-one correspondence between elements of \(X\) and elements of \(Y\). So a perfect matching is a bijection between \(X\) and \(Y\).
The symbol \([n]\) stands for the set of \(n\) distinct objects, probably first \(n\) natural numbers.

As mentioned above, a graph can have a perfect matching only if it is of even order. This is true for complete graphs also. So a complete graph has a perfect matching only if it is of even order. Fortunately, this is sufficient also – a complete graph has a perfect matching if and only if it is of even order. To find a matching in a complete graph of even order, consider a spanning cycle and delete alternate edges.

Computation of the number of perfect matchings in \(K_{2n}\) is given in page 108 of the Textbook DW [example 3.1.3].

Even though the graph shown above is of even order (6), it contains no perfect matching. A perfect matching in this graph needs three edges, to saturate six vertices. But, each of the vertices \(x\) and \(w\) is common to three edges. Since the graph contains five edges only, in any set of three or more edges in this graph, at least two of must share either \(x\) or \(w\). So, a matching in this graph can have at most two edges only. This justifies the claim.

By now you must have understood the concept of matching.

We now consider more examples

**Example 3:** Let us consider the graph given in the next page (see Fig.5).

In the graph we observe that \(M_1=\{e_1, e_3, e_5, e_{15}\}\), \(M_2=\{e_2, e_6, e_{10}, e_{11}\}\), \(M_3=\{e_7, e_{14}\}\) and \(M_4=\{e_2, e_4, e_6\}\) are matchings.
You can also observe that none of these matchings is perfect matching. This is a graph of even order 10. A perfect matching in this graph must contain five edges. $M_1$ and $M_2$ contains four edges only and $M_3$ contains two only. It is possible to find more matchings of size four, possibly not disjoint from already we have. You can try to find a perfect matching by adding any one of the remaining edges in the graph. But certainly fail. So this graph also has no perfect matching. Try to find a formal proof.

\[\square\square\square\]

Note that the above figure is a re-drawing of the Petersen graph. [See page 13 of Textbook DW for a formal definition of the Petersen graph and page 17 exercise 1.1.24 for different drawings.]

You know that an edge can be represented by its own label or using the labels of its end vertices. In the following examples we adopt the later.

**Example 4:** Let us consider the following graph.

Consider the graph represented by the diagram shown above. Then $M_1 = \{v_1v_2, v_3v_4\}$, $M_2 = \{v_1v_2, v_6v_3, v_4v_5\}$ and $M_3 = \{v_1v_6, v_2v_3, v_4v_5\}$ are matchings. $M_1$ is not perfect [why?]. Both $M_2$ and $M_3$ are perfect.

From this example we can observe that a graph can have more than one perfect matchings.
Try these exercises now

E1) Label the edges of the graph given in the figure below, prepare a list of matchings and identify perfect matchings, if any.

![Fig.7](image)

E2) Label the vertices of the graph given in the figure below, prepare a list of matchings and identify perfect matchings, if any.

![Fig.8](image)

From examples above, we have seen that a graph can have more than one perfect matchings. A perfect matching has to saturate all the vertices in the graph and hence every perfect matching must of size equal to half of the number of vertices in the graph. So, perfect matchings can exist only in graphs of even order. However it is not a sufficient condition as we have seen in note (viii) above. Moreover a proper subset of a perfect matching is not at all a perfect matching. But it is true that every subset of a matching is a matching.

From Example 1, it is clear that two matchings in a graph may be disjoint or may have some edges in common. $M_1$ and $M_3$ are disjoint, $M_2$ and $M_4$ have the edges $e_2$ and $e_6$ in common. It is possible to obtain new matchings from $M_3$ and $M_5$ by providing more edges. But this is not possible with $M_1$ and $M_2$. This lead to the concepts of maximal matchings. But there may be maximal matchings of different sizes as we see later. This leads to the concept of maximum matchings. We discuss these two concepts in the next section.
Maximal matching, maximum matching are two optimal concept resulting from the comparison of the matchings in graph with respect to set inclusion and size of matchings.

Let us start with the formal definitions.

**Definition 1:** A matching $M$ in a graph $G$ is called a *maximal matching* if it is not a proper subset of any other matching in $G$.

**Definition 2:** A matching $M$ in a graph $G$ is called a *maximum matching* if there is no matching in $G$ of bigger size.

**Definition 3:** Let $M$ be a matching in a graph $G$. Then a path $P$ in the graph $G$ is an $M$-alternating path if the edges of $G$ appearing along $P$ are alternatively in $M$ and not in $M$.

**Definition 4:** A path $P$ is $M$-augmenting if $P$ is $M$-alternating and both the end vertices of $P$ are not saturated by $M$.

Consider the Petersen graph given in Fig. 3. Sec 6.2. $M_1=\{e_1, e_3, e_{10}\}$ is a matching. But it is not a maximal matching since the $M_2=\{e_1, e_3, e_{10}, e_{15}\}$ is a matching containing $M_1$. Now try to include more edges to $M_2$. The edges $e_2$ and $e_4$ are sharing end vertices with $e_3 \in M_2$; $e_5$, $e_6$ and $e_7$ are sharing end vertices with $e_1$; $e_8$ and $e_9$ are sharing with $e_3$; $e_{11}$ is sharing with $e_{10}$; $e_{12}$ and $e_{13}$ are sharing with $e_{15}$ and sharing with $e_{10}$. So $M_2$ is maximal matching. Similarly you can find more maximal matchings starting with $M_1$ and choosing an edge other than $e_{15}$ to include. Consider $M_3=\{e_1, e_3, e_{10}, e_{12}\}$. This is not maximal since $M_4=\{e_1, e_3, e_{10}, e_{13}\}$ is a matching containing it. No more edge can include since the graph has only ten vertices. So this is also a maximal matching. But as already observed, size of a matching in this graph will not exceed five. So this is a matching of maximum size – a maximal matching. Moreover this is a perfect matching also.

Consider again the matching $M=\{e_1, e_3, e_{10}\}$. The paths $P_1=v_1v_2v_3v_4v_5$, $P_2=v_6v_1v_2v_3v_4v_5$ and $P_3=v_6v_1v_2v_3v_4v_3v_10v_7v_5$ are $M$-alternating paths, $P_3$ is an $M$-augmenting path while $P_1$ and $P_2$ are not since the vertex $v_5$ is $M$-saturated due to $e_{10}$ in $M$. $P_4=v_6v_1v_2v_3v_4v_5v_{10}v_7v_9$ is not an $M$-alternating path since $e_{11}$ and $e_{13}$ are not in $M$. 

![Fig.9](image-url)
NOTES:

(i) **Page 108 lines 21 to 23**

As mentioned earlier, maximal is related to set inclusion and maximum is related to size of the matching. Maximum matchings are maximal also, since no matching of bigger size exists in the graph. But, maximal need not be maximum. A perfect matching is always maximum and hence maximal.

Consider from matchings given by \( M_1 = \{ v_1v_2, v_3v_4 \} \), \( M_2 = \{ v_1v_2, v_3v_6, v_4v_5 \} \), \( M_3 = \{ v_1v_6, v_2v_3, v_4v_8 \} \) and \( M_4 = \{ v_2v_6, v_3v_5 \} \) in the graph whose diagram is given. Here both \( M_2 \) and \( M_3 \) are maximum as well as maximal. \( M_4 \) is a maximal matching since it is impossible to find another matching in \( G \) properly containing \( M_4 \) but it is not a maximum matching. \( M_1 \) is not maximal since \( M_5 = \{ v_1v_2, v_3v_4, v_5v_6 \} \) is a matching in \( G \) properly containing \( M_1 \).

(ii) **Page 109 lines 3 to 5**

In \( M \)-augmenting path, both end vertices are \( M \)-unsaturated. So the pendent edges are not in \( M \) and alternate edges are in \( M \). An immediate consequence is that the length of an \( M \)-augmenting path is odd.

The concept of augmenting path is helpful in checking maximum matching and in obtaining a matching of bigger size from a matching, if possible.

In the above figure, consider the matching \( M = \{ v_1v_2, v_3v_4 \} \).
The path $v_1v_2v_3v_4v_5$ is an M-alternating path and not an M-augmenting path since the end vertex $v_1$ is M-saturated. The path $v_6v_1v_2v_3v_4v_5$ is an M-augmenting path since it is M-alternating and both its end vertices are M-unsaturated.

(iii) **Page 109 lines 14 and 15**

The symmetric difference $A\Delta B$ of two sets $A$ and $B$ is given by $A\Delta B = (A \cup B) - (A \cap B)$ and also by $A\Delta B = (A - B') \cup (B - A')$ where the prime (') stands for set theoretic complementation.

(iv) **Page 109 lines 23 and 24**

This lemma is used to prove theorem 3.1.10 page 109 of *Textbook DW*.

(v) **Page 109 lines 14 and 15**

Theorem 3.1.10 characterises maximum matching in terms of augmenting paths.

Let us consider more examples

**Example 5:** Let us consider the following graph.

![Fig.11](image1)

Consider the matchings $M_1=\{e_1, e_7\}$, $M_2=\{e_1, e_3, e_{10}\}$ in this tree. Both $M_1$ and $M_2$ are maximal and $M_2$ is maximum. There is no perfect matching in this graph [why?]. The path $P_1=v_{10}v_8v_3v_4v_5$ is an $M_1$-augmenting path but not $M_2$-augmenting.

**Example 6:** Let us consider the following graph.

![Fig.12](image2)
In the figure consider a matching \( M = \{ v_1v_3, v_4v_6, v_5v_9, v_8v_{11} \} \). Then the path
\( P_1 = v_2v_4v_6v_9v_5v_8v_{11}v_7 \) is an \( M \)-augmenting path. But
\( P_2 = v_2v_4v_6v_9v_8v_7 \) is not \( M \)-augmenting since it is not an \( M \)-alternating path.

Try to do the following exercises:

E3) If \( M \) is a perfect matching in a graph \( G \), then is it possible to have an
\( M \)-augmenting path in the graph? Justify your answer.

E4) Determine the minimum size of a maximal matching and size of a
maximum matching in the cycle \( C_n \).

Remark 1: Note that the concept of maximal matching is a set theoretic
property in the sense that there is no 'bigger' matching in the graph while that
of maximum matching is related to the size in the sense that the graph contains
no matching of bigger size.

Remark 2: You can note that a graph may have more than one maximum
matchings and more than one maximal matchings. All maximum matchings
will be of same size but maximal matchings may differ in size. Further, note
that every perfect matching is maximum as well as maximal and every
maximum matching is maximal. A maximal matching need not be a maximum
matching. If there exist a perfect matching in a graph \( G \), then maximum
matchings are precisely the perfect matchings. So if you have a maximum
matching in a graph and if it is not perfect then the graph has no perfect
matchings. This idea also will help in checking and finding perfect matchings.

Theorem 3.1.10 in page 109 of the textbook DW is helpful in checking
maximum matchings.

We summarise the above discussion by highlighting the following points:

- a subset of a matching is also matching
- a graph can have more than one maximal matching
- maximal matchings in a graph may be of different size
- a graph can have more than one maximum matchings
- all maximum matchings in a graph must be of same size
- a maximum matching is maximal also
- a maximal matching need not be maximum
- every non-empty graph have a non-empty matching
- a graph need not have a perfect matching
- a perfect matching, if exists, is maximum and hence maximal as well
Under the terminology we studied in this section, the assignment problem is to find a matching in the graph model, that saturates every vertex in the partite set $A$. Further we look at some more conditions of the graph model, as given below. We note that

a) the number of applicants must not be less than the number of job positions, i.e. $|B| \geq |A|$

b) there must be at least one candidate qualified for each job, i.e. every vertex in $A$ has non-zero degree.

c) for every collection of $k$ jobs there must be at least $k$ candidates qualified for those $k$ jobs. i.e. $|N(S)| \geq |S| \forall S \subseteq A$

Note that the first two conditions obviously follow from the third. In fact, in 1935 an English Mathematician Phillip Hall [1904-1982] proved that the condition ‘$|N(S)| \geq |S| \forall S \subseteq A$’ is sufficient for a graph to have the required matching. Thereafter the condition is called Hall’s Matching condition in honour of him.

This is discussed in the next section.

### 6.4 HALL’S MATCHING CONDITION

Here we shall illustrate the Hall’s Matching condition. We shall first look at the assignment problem. We have seen that the solution of the problem requires the existence of a matching that saturates every vertex in a partite set $X$ in a bipartite graph $G$ with partite sets $X$ and $Y$ with $|X| \leq |Y|$. This was answered by Hall. In this section we discuss the related theorem due to Hall.

You may read the portion marked in the box.

**READ THE BOOK DW chapter 3, section 1 FROM PAGE 110 line 10 to page 111 last line.**

**NOTES:**

(i) **Page 110, line 18**

The condition $|N(S)| \geq |S| \forall S \subseteq X$ is the Hall’s condition for the existence of a matching that saturates every vertex in $X$. 
Graph Theory

Even though this condition seems to be simple, every subset of the partite set $A$ in the graph $G$ must satisfy this condition. This is important when one use the condition for verification.

(ii) **Page 110, line 20**

Theorem 3.1.11 in text DW page 110 is called Hall’s marriage theorem when $|X|=|Y|$. [refer line 24 page 111 of *Textbook DW*].

(iii) **Page 110, lines 23 and 24**

In the sufficiency part, we have to prove existence of matching that saturates every vertex in $X$. Instead the contrapositive is proved in Theorem 3.1.11 *Textbook DW*.

The statement to be proved is that ‘if

\[
|N(S)| \geq |S| \forall S \subseteq X,
\]

then the $X,Y$-bipartite graph has a matching that saturates $X$'.

The contra positive of this statement is that ‘if an $X,Y$-bipartite graph does not have a matching that saturates $X$, then must exist a subset $S$ of $X$ such that

\[
|N(S)| < |S|.
\]

(iv) **Page 111, lines 24**

A classical problem called the *Marriage problem* is a special case of the assignment problem, in which the graph in the model has partite sets of equal size. This has been settled by Frobenius in 1917, may be without using graph theory.

**The marriage problem**

Consider a set of $n$ girls and a set of $n$ boys. Some of the girls love some of the boys and every girl has at least one boy whom she loves. Is it possible to find a one-to-one alliance among these boys and girls?

In the light of Hall’s condition such an alliance is possible only if there is set of $k$ boys to match with every set of $k$ girls. A graph model of this problem is a bipartite graph with the set of girls and the set of boys as partite sets and an edge between two vertices in these two sets if and only if the respective girl loves the boy represented by the other vertex. Now the question is about the existence of a perfect matching in the bipartite graph.

(v) **Page 111, lines 30**

A $k$-regular $X,Y$ – bigraph means a bipartite graph with partite sets $X$ and $Y$ with every vertex is of degree $k$. 

---

**If p and q are two statements, then the implication ‘p is true whenever q is’ is equivalent to ‘q is not true whenever p is not true’.

The latter implication is called the contrapositive of the former.

The method of contrapositive proof is common in mathematics.**
Example 7: Let us consider the following graph.

![Fig. 13](image)

For $S=\{x_1, x_2, x_3\}$ in the figure above $N(S)=\{y_1, y_2\}$. So $|N(S)| < |S|$ and hence by Hall’s theorem this graph has no matching that saturates the partite set $X=\{x_1, x_2, x_3, x_4\}$.

Example 8: Let us consider the following graph.

![Fig. 14](image)

In this graph $M=\{x_1y_1, x_2y_3, x_3y_4, x_4y_2\}$ is a matching that saturates $X=\{x_1, x_2, x_3, x_4\}$. You can verify Hall’s condition for every subset of $X$.

Here are some exercises for you.

E5) Let $M$ and $N$ be matchings in an $X,Y$-bigraph $G$. Suppose $M$ saturates a subset $S$ of $X$ and $N$ saturates a subset $T$ of $Y$. Prove that $G$ contain a matching that saturates $S \cup T$.

E6) Try the above exercise when $G$ is any graph and $S$ and $T$ are any two disjoint subsets of $V(G)$.

E7) Try the above exercise when $G$ is any graph and $S$ and $T$ are any two subsets of $V(G)$.

From the Hall’s theorem, it is clear that the assignment problem and its special case, the marriage problem, have a solution if and only if the condition $|N(S)| \geq |S| \forall S \subseteq X$ holds. There are algorithms to find out such an assignment [matching]. We are not included this here as it is not part of the syllabus here. Some algorithms are discussed in Section 3.2 of Textbook DW.

### 6.5 A MIN-MAX THEOREM

In this section we shall discuss the min-max theorem is, due to Konig and Egervary, which establishes the equality of the size of maximum matching and minimum vertex cover in any bipartite graph. There are other min-max theorems elsewhere.
NOTES:

(i) Page 112 lines 10 and 11

A vertex cover of a graph $G$ is a subset of $V(G)$ such that it contains at least one end vertex of each edge in $G$. Note that any subset of $V(G)$ that contains a vertex cover is also a vertex cover and trivially $V(G)$ itself is a vertex cover of $G$.

The following figure illustrates this.

![Figure 15](image)

In figure $A = \{v_1\}$ and $B = \{v_2, v_3, v_4, v_5, v_6, v_7\}$ are vertex covers of $G$. Any set of vertices containing $v_1$ is a vertex cover however no subset of $B$ is a vertex cover.

Any set of $n-1$ vertices in $K_{n,n}$ is vertex cover but no subset of less vertices is a vertex cover since no such set will cover the edge joining the exempted vertices.

(ii) Page 112 line 19

If we look at Example 3.1.15 in page 112 textbook DW you will motivate to Theorem 3.1.16 page 112 textbook DW. In the figures, the vertices belonging to vertex cover are marked by enclosing them in squares and the edges are the matching are highlighted with bold face.

(iii) Page 112 line 25

The theorem assures the existence of a vertex cover same as in size that of a maximum matching and a matching of size same as that of a minimum vertex cover in a bipartite graph. The proof is by exhibiting the required sets assuming the hypothesis [maximum matching / minimum vertex cover]. This theorem used in a proof of an important theorem, Menger’s theorem [page 167 Textbook DW]. Theorem 3.1.16 ,page 112, Textbook DW hold good in bipartite graphs only and not true in general graphs.
The second graph in the figure for Example 3.1.15 page 112 of the Textbook DW is a counter example. In the graph a vertex cover need at least three edges since every vertex is of degree 2, two vertices can cover only four edges but the graph has five edges. A maximum matching contains only two edges since for having a matching with three edges it need at least six vertices in the graph.

Now we see an example:

Example 9: Let us consider the following graph.

Here \( \{x_1, y_3, x_3, y_1, x_4, y_4\} \) is a maximum matching \( \{x_1, x_4, y_3\} \) is a minimum vertex cover. Since the given graph is bipartite, by Theorem 3.1.16, the maximum size of a matching equals minimum size of a vertex cover of the graph.

We have seen that a matching \( M \) is a subset of the edge set whose elements have no common end vertex and a vertex cover is subset \( S \) of the vertex set such that every vertex in the graph is adjacent to some vertex in \( S \). Recall the concept of independent sets you have studied in Unit 1, an independent set is a subset of the vertex set such that no pair of them form an edge in the graph. This is dual to the concept of vertex cover. In comparison with this, a matching can be considered as an independent set of edges. Then a natural question of an edge cover arises. This is also of interest in graph theory and its applications. The concepts of independent set, edge cover, some of the relations between them and graph parameter related to there are discussed in the next section.

6.6 INDEPENDENT SETS AND COVERS

Recall that a subset \( S \) of the vertex set \( V(G) \) of a graph \( G \) is independent if the subgraph induced by \( S \) is an empty graph, i.e. no two vertices in \( S \) form an edge in \( G \). This is similar to the concept of matching since matching is a subset of \( E(G) \) such that no pair shares a vertex and independent set a subset of \( V(G) \) without sharing of edges. One can view a matching as an independent set of edges. Edge cover is a concept similar to vertex cover.

You will learn about these concepts after reading the relevant portions of the Textbook DW mentioned in the box given in the next page.
Graph Theory

NOTES:

(i) Page 113 second line from bottom

Recall the definition of independent set in a graph; refer Textbook DW page 4 lines 3 and 4. This concept is dual to the concept of vertex cover defined in the previous section; refer Textbook DW page 112 line 9. An independent set in a graph G is a subset S of the vertex set V(G) such that no edge in G joins any pair of vertices in S. Note that matching can be considered an independent set of edges. Naturally, any subset of an independent set is also an independent set. Hence, as in matching, the questions of maximal and maximum independent sets arise.

(ii) Page 113 last two lines from bottom and page 114 line 16

The independence number $\alpha(G)$ of a graph G is the maximum size of an independent set in G. Convince yourself that independence number of any complete graph is 1 since each vertex is adjacent to every other vertex. Independence number of a complete bipartite graph is the cardinality of the bigger partite set. Independence number of an even cycle of length $2n$ is $n$ and that of an odd length $2n+1$ is also $n$.

Let G be the Petersen graph shown above. Let $S_1 = \{v_1, v_3, v_7\}$. Because the presence of $v_1$ forbids $v_2$, $v_5$ and $v_6$ from $S_1$, being adjacent to it, presence of $v_3$ forbids $v_2$, $v_4$ and $v_8$ and $v_7$ forbids $v_2$, $v_9$ and $v_{10}$. But $S_1$ is not a maximum independent set since $S_2 = \{v_1, v_3, v_9, v_{10}\}$ is an independent set containing more elements. But G contains no independent set containing more than four elements. Since an independent set in G can have only at most two vertices each from the cycles $v_1v_2v_3v_4v_5v_1$ and $v_6v_9v_7v_{10}v_8v_6$.

So the independence number of the Petersen graph is 4.
(iii) Page 114 line 1

The graph in the figure is bipartite but not complete and the observation in the example after note (ii) is not contrary to that in Example 3.1.18 page 114 Textbook DW.

(iv) Page 114 line 6

The concept of edge cover is the counterpart of vertex cover.

Moreover, if L is an edge cover of a graph G then every vertex in G is an end vertex of some edge in L. So every edge in G must share end vertices with at least one edge in L. Hence concept of edge cover can be considered as a dual to the concept of matching. An edge cover of a graph of order n must contain at least $\left\lfloor \frac{n}{2} \right\rfloor$ edges.

where $\lfloor x \rfloor$ stands for the least integer greater than or equal to n.

Any sub set of E(G) containing an edge cover of G is also an edge cover of G. This motivates the concept of minimum edge cover.

(v) Page 114 lines 11 and 12

A perfect matching in a graph contains one edge each adjacent to every vertex and hence is an edge cover.

(vi) Sufficient explanation is given in textbook DW related to definition 3.1.20 and matter thereafter in page 114 and 115

The following example provides an illustration.

Example 10: Let us consider the following graph.

![Fig.18](image)

In the graph shown above {$v_1$} and {$v_3$} are independent sets. The set {$v_1$} is a maximal independent set but not the set {$v_3$}. Also the set {$v_3, v_5, v_7$} is a maximum independent set and the set {$v_1v_2, v_1v_3, v_1v_4, v_1v_5, v_1v_6, v_1v_7$} is an edge cover of G. The set {$v_1v_2, v_3v_4, v_5v_6, v_7v_1$} is also an edge cover of G.
This is a minimum edge cover of $G$. An edge cover of this graph must be of size at least four, since $G$ has seven vertices.

With this we come to an end of our discussion on independent sets and covers and also to this unit.

Let us now summaries the points discussed in this unit.

### 6.7 SUMMARY

In this unit, we have covered the following:

1. The concepts of matching, perfect matching, maximal matching, maximum matching, vertex saturated by a matching, alternating and augmenting paths, vertex cover, edge cover and independent set.

2. The number of perfect matchings of $K_{n,n}$ is $n!$

3. The number of perfect matchings of $K_n$ is $(2n)!/2^n n!$

4. A maximal matching need not be a maximum matching.

5. A perfect matching is maximal and maximum also.

6. Every component of the symmetric difference of two matchings is a path or an even cycle.

7. A matching $M$ in a graph $G$ is a Maximum matching in $G$ if and only if $G$ has no $M$-augmenting path.

8. Berge’s characterization of a maximum matching in terms of augmenting path.

9. If $(X, Y)$ is a bipartite graph with $|X| \leq |Y|$, a necessary and sufficient condition for the existence of a matching $M$ that saturates all vertices of $X$ is the Hall’s condition.

10. Every $k$-regular bipartite graph has a perfect matching.

11. In a bipartite graph, the maximum size of a matching equals minimum size of a vertex cover.

12. Duality of matching and covering.

13. The four parameters $\alpha(G)$, the maximum size of independent set in a graph $G$; $\alpha'(G)$, the maximum size of a matching in $G$; $\beta(G)$ minimum size of a vertex cover and $\beta'(G)$, the minimum size of an edge cover in $G$ and the following relations among these parameters:

   (i) $\alpha'(G) = \beta(G)$ if $G$ is bipartite,

   (ii) $\alpha(G) = \beta'(G)$ if $G$ is bipartite and without isolated vertices.
(iii) \( \alpha(G) + \beta(G) = n(G) \) in any graph

(iv) \( \alpha'(G) + \beta'(G) = n(G) \) if \( G \) has no isolated vertices

6.8 HINTS/SOLUTIONS

E1) Try it by yourself
E2) Try it by yourself
E3) No, Augmenting path need unsaturated vertices. But a perfect matching saturates every vertex.
E4) Consider \( C_n \) figure below (see Fig.19)

Label the edges of \( C_n \) consecutively by \( e_1, e_2, e_3, \ldots, e_n \) as in figure. Consider the matching \( M = \{ e_1, e_4, e_7, \ldots \} \) the last vertex in \( M \) is \( e_{n-2} \) if \( n \) is a multiple of 3, otherwise \( e_{n-1} \). Then \( M \) is a maximal matching since every other edge shares a common vertex with one edge in \( M \). However no set of less edges will be a maximum matching.

For this look at the following special cases.

Case 1: Consider \( C_4 \)

Then \( \{v_1v_2, v_3v_4\} \) is a maximal matching, but no set of one edge is maximal.

Case 2: Consider \( C_5 \)
Then \( \{v_1v_2, v_3v_4\} \) is a maximal matching but no set of one edge is maximal.

**Case 3:** Consider \( C_6 \)

Then \( \{v_1v_2, v_4v_5\} \) is a maximal matching but no set of one edge is maximal.

So, if \( n = 0 \) (mod 3), then the minimum size of a maximal matching = \( n/3 \). If \( n = 1 \) (mod 3) or \( 2 \) (mod 3), then the minimum size of a maximal matching = \( \left\lceil \frac{n}{3} \right\rceil + 1 \)

**E5)**  **Hint:** Consider some convenient subset of \( M \cup N \).

**E6)** Consider the situation in which an edge in \( M \) and an edge in \( N \) have a common end vertex \( u \notin S \cup T \).

**E7)** Same argument as in E6

---

### Additional Problems

**Problem 1:** For any graph \( G \), let \( \text{odd}(G) \) denote the number of odd components of \( G \). If \( G \) has a perfect matching then, prove that \( \text{odd}(G-S) \leq |S| \forall S \subseteq V(G) \).

**Solution:** Let \( G \) have a perfect matching \( M \) Let \( S \subseteq V(G) \)

Let \( \text{odd}(G-S) = k \) say \( G_1, G_2, \ldots, G_k \). Since each \( G_i, i = 1 \) to \( k \) has odd number of vertices, at least one vertex of each \( G_i \) must be matched to some vertex of \( S \) in the matching \( M \).

So, \( S \) must have at least \( k \) vertices i.e. \( |S| \geq \text{odd}(G-S) \forall S \subseteq V(G) \).

**Problem 2:** Check whether the following graph has a perfect matching.
**Problem 3:** Let four girls a, b, c and d know 3 boys w, x, y and z as given below.

<table>
<thead>
<tr>
<th>Girl</th>
<th>Boys known</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>w</td>
</tr>
<tr>
<td>b</td>
<td>x</td>
</tr>
<tr>
<td>c</td>
<td>y</td>
</tr>
<tr>
<td>d</td>
<td>z</td>
</tr>
<tr>
<td></td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>z</td>
</tr>
<tr>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

Draw the bipartite graph and check Hall’s condition for this problem.

**Hint:** You have to check Hall’s condition for all 15 non-empty sub sets of \{a, b, c, d\} and if the condition fails for one no preferred matching is possible.

**Problem 4:** Check whether the bipartite graph given in the following figure has a matching which saturates all the vertices of X.

![Fig.24](image)

**Problem 5:** A school wants to fill six vacancies of teachers with one teacher each for Mathematics, Physics, Chemistry, Biology, Computer Science and Yoga. In order for a teacher to be selected for a particular subject, he or she must have majored or minored in that subject. The school received six applications from six candidates A, B, C, D, E and F with the following qualifications.

- A: major Physics, minor Mathematics
- B: major Biology, minor Chemistry
- C: major Chemistry, minor Yoga
- D: major Mathematics, minor Computer Science and Yoga
- E: major Computer Science, minor Mathematics
- F: major Physics, minor Mathematics and Chemistry

What is the maximum number of applicants the school can hire?

**Hint:** Draw a bipartite graph with partite set X representing the six subjects, partite set Y representing the six candidates and an edge joins a vertex x in X to a vertex y in Y if the candidate x is qualified for a post in the subject y. Then check the Hall’s condition for all non-empty subsets of X.
7.1 INTRODUCTION

The connectivity of a graph is a concept having much application in network theory. We have already discussed the concept of connectedness [definition 1.2.6 especially in communication network page 21, Textbook DW]. Connectedness is a structural property while connectivity is a measure of the strength of connectedness. Using connectivity of graphs it is possible to analyze the parameters of a network related to its strength, capacity, reliability, efficiency, etc. To minimize the cost, the number of links must be minimized and for efficiency numbers of transit points are to be minimised. But the system should be so designed that even if one or two vertices (transit points / nodes) fail, the communication is not disrupt. Similarly, even if one or two edges (cable/links) fail the communication should remain unaffected.

For any graph G two graph invariants namely connectivity (vertex connectivity), and edge connectivity are discussed here. These parameters are helpful in network analysis. An important theorem due to Menger, concepts of blocks, k-connected graphs, k-edge connected graphs and some concepts and results in network analysis are also discussed.

Knowledge of Graph theory discussed so far, a little knowledge of weighted digraphs and knowledge of the pigeonhole principle are prerequisites for this unit. The contents of sections 1.4 and 2.3 of Textbook DW will provide necessary information on digraphs and weighted graphs respectively.

In this unit, also we restrict to simple graphs unless otherwise stated.

Objectives

After studying this unit, you should be able to

- identify vertex cuts, edge cuts, disconnecting sets in a graph;
- evaluate the vertex connectivity and edge connectivity of a graph;
- recognize k-connected graphs, and k-edge connected graphs;
blocks in a graph;
- obtain appropriate graph models of networks and analyze / check the system for optimality.

7.2 CONNECTIVITY

We have already seen the concepts of cut vertices and cut edges in the second unit [refer definition 1.2.12 page 23 Textbook DW]. But there are graphs without cut vertices or cut edges. Then, naturally, we will try to remove more vertices/edges to get the graph disconnected. This results in the question: what is minimum number of vertices/edges to be removed to get a disconnected graph. This leads to the concepts of (vertex) connectivity and edge connectivity.

In this section we discuss the vertex connectivity which is simply called connectivity.

Let G be a connected graph with vertex set V(G). A subset S of V(G) is called a vertex cut if its removal from G results in a disconnected graph or trivial graph. The minimum cardinality of such a set in a connected graph is called the connectivity of the graph.

You will learn more about this while reading the corresponding portion in the Text book D.W.

READ DW Page 149 line 1 to page 150 line 32.

NOTES:

(i) Page 149 line 8 :

It is sufficient to discuss connectivity in the context of simple graphs only since it is related to deletion of vertices.

(ii) Page 149 lines 9 to 12 :

Note that any superset of a vertex cut is also a vertex cut. So the minimum number of vertices in a vertex cut is important. The Connectivity \( \kappa(G) \) of a graph G is defined as the minimum cardinality of a vertex cut. That is, a graph G is of connectivity k if it has a vertex cut of order k and no vertex cut of order k – 1 or less. A graph of connectivity k or more is called k-connected.

By the definition, a k-connected graph is (k-1)-connected, (k-2)-connected, 2-connected, and 1-connected.

(iii) Page 149 lines 14 to 15 :

The connectivity of a graph is defined as the minimum number of vertices in a vertex cut.
Connectivity and Networks

(iv) Page 149 lines 16 to 12:

Recall that a complete graph is a clique. In the definition of vertex-cut, we have considered the removal of a set of vertices to result in a disconnected graph. According to this, a complete graph has no vertex cut. To incorporate complete graphs under the concept of connectivity, we consider an additional possibility of resulting in a trivial graph – a graph containing one vertex only. With this any subset of \( n-1 \) vertices in a complete graph is a vertex cut and no set of less vertices. This is justified by the discussion in Example 4.1.2 in Textbook DW. Due to this, the connectivity of a complete graph of order \( n \) is \( n-1 \).

For completeness of the theory, the connectivity of a disconnected graph and that of the trivial graph are defined to be zero.

(v) Page 150 lines 1 to 5:

To obtain a disconnected graph from \( K_{m,n} \), all vertices in one of the partite sets is to be removed. So, the connectivity of \( K_{m,n} \) is the minimum of the cardinality of the partite sets.

(vi) Page 150 lines 8 to 10:

We already mentioned this in note (iv) here.

(vii) Page 150 line 11:

The graphs considered in Example 4.1.3 of Textbook DW is of interest in combinatorics. It is defined and discussed in detail in 1.3.7 and 1.7.8 page 35 and 36 of Textbook DW.

Example 1:

Consider the graph in the following figure

![Graph](image-url)

Fig.1
In this graph, the sets \( \{ v_2, v_3, v_4 \} \) and \( \{ v_5, v_6 \} \) are vertex cuts but has no vertex cut with only one vertex [i.e. no cut vertex]. The connectivity of \( G \) is the minimum cardinality of a vertex cut which is 2 here. So this graph is 2-connected and 1-connected but not 3-connected.

Example 2: Let us show the connectivity of the hyper cube \( Q_k \) is \( k \).

We first note that \( Q_1 = K_2 \), \( Q_2 = K_2 \times K_2 \)
\( Q_k = Q_{k-1} \times K_2 \), \( k = 2,3,... \)
Let \( K_2 \) be labelled with 0 and 1.

We prove the result by induction on \( k \). if \( k = 1, \ 2 \), then the connectivity is \( k \). Assume it is true for all graphs with \( k \leq m-1 \) and Let \( k = m \geq 2 \).

Since each vertex \( Q_m \) is of degree \( m \), we can see that if we remove all vertices adjacent to a vertex \( u \), then is the graph gets disconnected. Hence the connectivity is \( \kappa(Q_m) \leq m \).

To prove \( \kappa(Q_m) = m \). It is enough to show that if \( S \subset Q_m \), then \( Q_m - S \) is disconnected. Then \( |S| \geq m \).

By induction hypothesis, \( \kappa(Q_1) = 1 \) \( \kappa(Q_2) = 2 \) \( \kappa(Q_{m-1}) = m-1 \)

Also \( Q_m = Q_{m-1} \times K_2 \). That is \( Q_m \) is (say \( Q \) and \( Q^1 \)) obtained by taking two copies of \( Q_{m-1} \) and by joining the corresponding vertices of \( Q \) and \( Q^1 \). Let \( S \) be a vertex cut of \( Q_m \). Then \( Q_m - S \) is disconnected then two carves area.

Case I: \( Q-S \) and \( Q'-S \) are connected. Since \( Q-S \) and \( Q'-S \) are connected but \( Q_m-S \) is disconnected, we can see that while deleting \( S \), all edges joining \( Q \) and \( Q^1 \) are deleted. That is \( S \) should contain atleast one end point of each of such edges i.e. \( |S| \geq 2^{m-1} \) but if \( m \geq 2 \), then \( m \leq 2^{m-1} \). Hence, \( |S| \geq m \).

Case II: \( Q-S \) or \( Q'-S \) is disconnected.

Suppose \( Q'-S \) is disconnected. Similar is the case when \( Q-S \) is disconnected. By induction hypothesis, \( |S \cap Q| \geq m-1 \). If \( S \) contains no vertex of \( Q' \), then \( Q' \) is connected and each vertex in \( Q'-S \) is connected to the
corresponding vertex in $Q'$. Hence $Q_{m\setminus S}$ is connected. A contradiction. Hence $S$ contains at least one vertex of $Q'$. Therefore $|S| \geq m$. So we have $\kappa(Q_m) = m$. Hence for any $\kappa$, $\kappa(Q_k) = \kappa$.

\[ \square \square \square \]

**Example 3:** If $C$ is a cycle $C$, then $k(C) = 2$ for any tree $T$ we have $(T) \kappa(T) = 1$.

Try these exercises now.

---

E1) Find the connectivity of the following Graph $G$.

![Graph G](image)

**Fig.3**

E2) What is the connectivity of a wheel $W_{1,n}$ for $n \geq 4$?

E3) Draw any regular graph with at least 5 vertices with connectivity 1.

As in the case of vertices, the removal of some edges from a connected graph may also result in a disconnected graph. This leads to the concept of edge connectivity and studied extensively and has applications also. We discuss this in the next section. For this the concept of disconnecting set corresponding to separating set and edge cut corresponding to vertex cut are considered.

### 7.3 EDGE CONNECTIVITY

Edge connectivity is a concept similar to connectivity. Here the separating set contains edges rather than vertices. Naturally, the question of minimum number of edges to be removed to obtain a disconnected graph arises and the result is the parameter edge connectivity $\kappa'(G)$. Actually this parameter is more useful in the analysis of a network system.

You can start reading the corresponding portions from the **Textbook DW**.

**READ DW** Page 152 line 1 to page 154 line 10.

---

**NOTES:**
Disconnecting set is set of edges whose removal disconnects the graph. Concepts of disconnecting set, edge connectivity and k-edge connected graphs are similar concepts of separating set [vertex cut], connectivity [vertex connectivity] and k-connected graphs.

You may specially note the symbol \([S,T]\). This denotes the set of all edges in a graph \(G\) with one end in \(S\) and other in \(T\) where \(S\) and \(T\) are any two disjoint subsets of \(V(G)\). Note that \(S\) and \(T\) are set of vertices while \([S,T]\) is a set of edges with end vertices in \(S \cup T\). So \(S \cup T\) together with \([S,T]\) forms a sub-graph of \(G\). Moreover it is bipartite too.

To define an edge cut in a graph \(G\), we consider any subset \(S\) of \(V(G)\) and its complement \(T = V(G) - S = \overline{S}\). Then \([S,T]\) = \([S, \overline{S}\] is a vertex cut. Note that every edge cut is a disconnecting set since the removal of an edge cut results in a disconnected graph. But the converse is not true.

The figure illustrates the difference between disconnecting set and edge cut. It shows a disconnecting set and an edge cut in the same graph. In the diagram on left, it is not possible to take a set of vertices such that all of the boldfaced edges have one end in \(S\) and other end in \(S\). The obstacle is the presence of (bold faced) edges in the disconnecting set incident at all vertices of an odd cycle [two three cycles and two five cycles]. Note the last observation in note (ii) here and the characterisation of bipartite graph in theorem 1.2.18 in page 25 of Textbook DW.

In a graph \(G\), any subset of \(E(G)\) containing a disconnecting set is also a disconnecting set. So the question of minimal disconnecting set arises. Smaller a smaller disconnecting set can be obtained by leaving suitable edges from disconnecting sets. Clearly this will end up in an edge cut after which it will not be possible to leave edges to retain a disconnecting set. So edge cuts are minimal disconnecting sets. This shows that very disconnecting set contains an edge cut. But the figure discussed in the earlier item note (iv), shows that a disconnecting set can contain more than one disconnecting set.
Theorem 4.1.9 is an interesting and useful relationship between the connectivity parameters discussed here and the minimum degree.

Example 4.1.10 gives a graph in which strict inequalities hold in the relationships in Theorem 4.1.9 on page 149 Textbook DW.

In a complete graph of order $n$ each of the parameters is equal to $n - 1$. In any cycle, each of these three parameters is 2.

Theorem 4.1.11 gives a class of graphs in which the equality holds for the first inequality in 4.1.9. A three regular graphs is one with every vertex is of degree three.

Example 4: Consider the graph given in the following figure.

![Fig.4](image)

Here, $\delta(G) = 3$, and $k(G) = k^{1-g} = 1$.

Example 5: The following figure illustrates that there are graphs with $\Delta(G) \geq 4$. See, the following figure.

![Fig.5](image)

Here, $\delta(G) = 3, k(G) = 1$ and $k^{1}(G) = 2$ and $\{u\}$ is vertex cut (or we may say $u$ is a cut vertex) and $\{e_1, e_2\}$ is a minimum edge cut.

□□□

Now you try the following problems.

E4) Prove that if $k(G) < k^{1}(G)$, then $\Delta(G) \geq 4$.

E5) Give an example of a graph with $k(G) = 1, k^{1}(G) = 2$ and $\delta(G) = 4$

E6) Draw a graph $G \neq K_5$ with $k(G) = k^{1}(G) = \delta(G) = 4$. 
We have seen that 2-connected graphs are graphs without cut vertices. But the complete graphs $K_1$ and $K_2$ are not 2-connected and have no cut vertex also. Absence of a cut vertex in a network increases its reliability. Thus, the class of graphs without cut vertices play prominent role in many contexts. We will study this class of graphs in the next section.

### 7.4 BLOCKS

Blocks are little stronger subgraphs in the sense that they are free from cut vertices of their own. Moreover they form a useful and natural decomposition of a graph into edge disjoint subgraphs.

**Definition 1:** A block of a graph is defined as a maximal connected subgraph without any cut vertex of its own.

The definition says that any connected proper supergraph of a block must contain a cut vertex of its own. Here the demand of cut vertex of its own is necessary since a block in a graph $G$ may contain a cut vertex of $G$.

In addition to this connected graph without cut vertex is also called block.

You will learn more above this when you start reading *Textbook DW*.

READ DW Page 155 line 11 to page 156 line 18.

**NOTES:**

(i) *Page 155 lines 13 to 15*

Neither a subgraph or a supergraph of a block is a block. Even though a block $B$ of a graph $G$ contains no cut vertex of its own, it will contain cut vertices of $G$ if [and only if] $B$ is a proper subgraph of $G$.

(ii) *Page 155 lines 16 to 17 :*

A block with two vertices is $K_2$ and with three vertices is $K_3$. But, a block with four or more vertices need not be a clique since cycles may turn out as blocks. These are clarified in example 4.1.17 *Textbook DW*.

**Example 6:** Consider the graph given in Fig 6(a)
Then the blocks are $B_1, B_2, B_3,$ and $B_4$ are given in the following figures.

![Diagram](https://example.com/diagram.png)

**Example 7:** If $T$ is a tree, any edges form a block in $T$.

**Note 1:** $K_2$ is a block if and only if it is not the part of any cycle in $G$.

**Note 2:** You may note that if $B_1$ and $B_2$ are two blocks in a graph $G$ which share a vertex $u$, then $u$ is a cut point in $G$.

Now, you try the following exercises.

- **E7)** Prove that every graph $G \neq K_2$ has fewer cut vertices than blocks.
- **E8)** Give an example of a graph $G$ with 5 blocks and (i) 1 cut point (ii) 2 cut points (iii) 3 cut points (iv) 4 cut points.
- **E9)** Let $G$ be a cubic graph with a cut point. Show that $G$ has at least two cut points.

A block with one vertex is $K_1$, which exists only in a disconnected graph with trivial components.

You have studied the definition of $k$-connected graphs in Definition 4.1.1 in page 149 of *Textbook DW*. But no more discussion was made. We discuss a special case, 2-connected graphs in the next section.

### 7.5 2-CONNECTED GRAPHS

The concept of $k$-connected graph is related to strength and reliability of a network. A network will not disrupt by the fault of one intermediate node if it...
Graph Theory

has alternative paths/line through which the flow can be retained or redirected. So it is better to have more disjoint paths (routes) connecting every pair of nodes. Installation of more routes between two nodes without using other nodes is not cost-effective. So we have to consider routing through intermediate nodes in such a way that every pair of terminals are connected by more than one disjoint rout. A network system corresponding to a 2-connected graph is a minimal one with this advantage.

The first part of this section discusses characterisations and properties of 2-connected graphs and then k-connected graphs.

A graph G is k-connected if for every subset S of V (G) with |S|<k, G – S is always connected i.e. every vertex cut contains at least k vertices.

You can note that a k-connected graph is always (k-1)-connected, (k-2)-connected, (k-3)-connected, . . . , 1-connected also. A graph with a cut–vertex is 1-connected but not 2-connected. A block with more than two vertices is two connected, but need not be 3-connected.

Now you can start reading the relevant portions in Textbook DW.

READ DW Page 161 line 1 to page 163 line 13

NOTES:

(i) Page 161 line 8

A graph is 2-connected if it has no cut vertex and hence has no cut edge also.

In a graph, if two paths joining two vertices have no common internal vertices then they are free from common edges also.

(ii) Page 161 lines 10 to 12

Here we give a proof of the theorem with the same arrangement as given in the Textbook DW.

The Theorem 4.2.2., in the Textbook DW due to Whitney assures the existence of two internally disjoint paths between any pair of vertices in a 2-connected graph. This contributes to measures of reliability.

(iii) Page 162 lines 1 to 3

The Expansion lemma (Lemma 4.2.3 page 162 textbook DW’) gives a tool to develop and extend a system of network without loosing its strength.

Here we give the proof in detail:
**Proof of Lemma 7.5.4 (Expansion lemma)**

Here we have to show that if $G$ is a $k$-connected graph and $G'$ is obtained by adding a new vertex say $y$ to $G'$ and joining it to at least $k$ vertices in $G$, then $G'$ is also $k$-connected.

The proof is done by showing that every separating set of $G'$ is of cardinality at least $k$.

Let us take an arbitrary separating $S$, in $G'$ we will show that $|S| \geq k$.

**Case I:** Let $y \in S$. Then $G-(S \setminus \{y\})$ is disconnected. i.e. $S \setminus \{y\}$ is a separating set in $G$ and hence $|S \setminus \{y\}| \geq k$ since $G$ being $k$-connected. So $|S| \geq k + 1 > k$.

**Case II:** $y \notin S$ and $N(y) \subseteq S$. Since $\deg(y) \geq k$ in $G'$, clearly $|S| \geq k$.

**Case III:** $y \notin S$ and $N(y) \not\subseteq S$. Then $N(y)$ contains an element $v$ which is not in $S$. Then $y$ and $v$ are adjacent in $G'-S$ and hence lie in the same component of $G'-S$. Let $u$ be a vertex in some other component of $G'-S$. Then $u$ and $v$ will be in different components of $G-S$. This shows that $G-S$ has more than one component and so $S$ is a disconnecting set of $G$ also. So, $|S| \geq k$.

Thus a disconnecting set of $G'$ always contains at least $k$ vertices. This proves the expansion lemma.

(iv) *Page 162 lines 8 to 9*

Theorem 4.2.4 page 162 Textbook DW gives three characterizations of 2-connected graphs.

(v) *Page 162 lines 2 to 1 from bottom*

The concept of subdivision given in Definition 4.2.5 page 162 of Textbook DW is important in the study of planarity of graphs [refer unit 9]. This concept plays a prominent role in characterization of planar graphs [refer Theorem 6.2.2 page 246 of Textbook DW, not included in your syllabus].

(vi) *Page 163 lines 1 to 2:*

Corollary 4.2.6 assures the retaining of the strength of the system even after the introduction of an intermediates node in a line/edge.
Let us see some examples.

**Example 8:** Consider the following graph.

![Graph](image)

In the graph shown above, consider the following paths connecting $u_1$ and $u_7$.

- $P_1: u_1 - u_2 - u_6 - u_7$
- $P_2: u_1 - u_4 - u_5 - u_7$
- $P_3: u_1 - u_2 - u_3 - u_6 - u_7$
- $P_4: u_1 - u_2 - u_3 - u_4 - u_8 - u_7$

Here $P_1$ and $P_2$ are internally disjoint. $P_2$ and $P_3$ are internally disjoint.

$P_2$ and $P_4$ are not internally disjoint.

**Example 9:** Let us consider the following graphs.

![Graphs](image)

Here $G'$ is obtained from $G$ by the subdivision the edge $u_2 - u_4$.

We discussed 2-connected graphs in this section. The general situation, $k$-connected graphs, remains untouched. The next section is devoted for that along with $k$-edge connected graphs.

### 7.6 k-CONNECTED AND k-EDGE CONNECTED GRAPHS

In this section we discuss $k$-connected graphs and $k$-edge connected graphs. This creates a background for an important theorem due to Menger which is helpful in the theory of networks.
So far we have focused on set of vertices/edges whose removal results in disconnected graphs. Now we insist on a localized situation of getting a disconnected graph such that two designated vertices are in different components of the resulting graph.

You can start reading Textbook DW.

READ DW Page 166 line 8 to page 168 line 18

NOTES:

(i) Page 166 lines 14 to 18:

These lines define two concepts x,y –cut and x,y-path. The former disconnects while the later connects.

If S is an x,y-cut in a graph G then any subset of V(G) − {x,y} containing S is also an x,y-cut. So the existence of a minimal x,y-cut came in.

On the other hand any subset of a set of x,y-paths is also a x,y-path. This is true for pairwise internally disjoint x,y-paths also. Then the question of a maximal set of pairwise internally disjoint x,y-paths arises. These two optimal sets contribute two local parameters \( \kappa(x, y) \) and \( \lambda(x, y) \).

(ii) Page 166 lines 20:

If x and y are two vertices in a graph, to avoid an x and y path, it is necessary to remove some vertex of that path from G. So an x,y-cut must contain at least one vertex from each x,y-path. So is a minimal x,y-cut. Hence \( \kappa(x, y) \) must not be less than the number of pairwise internally disjoint x,y-paths. In other words, the size of a maximal set of pairwise internally disjoint x,y-paths will not exceed \( \kappa(x, y) \). This results in the given inequality.

We shall make a remark here

Remark: To avoid an x,y-path in a graph, it is sufficient to remove an edge of the path. But here the definition of x,y-cut considers only vertices in it.

We shall now look at some examples.
**Example 10:** Let us consider the graph given below:

In this graph, \(\{u_1, u_2, u_3\}\) is a \(v_1 - v_2\) cut? Similarly \(\{v_1, v_2, v_3\}\) is a \(u_1 - u_2\) cut.

**Example 11:** Let us find \(k(x, y)\) for the following graph.

Here \(S = \{a_1, v_1, w_1, u_1, P_1\}\) is on \(x - y\) cut. Hence \(k(x, y) \leq 5\).

The \(x - y\) paths are

\[
\begin{align*}
P_1 &: x - w_1 - y \\
P_2 &: x - a_1 - a_2 - a_3 - a_4 - y \\
P_3 &: x - v_1 - y \\
P_4 &: x - u_1 y \\
P_5 &: x - P_1 - P_2 - y
\end{align*}
\]

These paths are internally disjoint, hence \(k(x, y) = 5\).

You can now start reading **Textbook DW.**

**READ DW Page 167 line 1 to page 168 line 10**

(iii) **Page 167 lines 1 to 6**

The remark given in these lines highlights the importance and celebrity of the so called Menger’s theorem.

(iv) **Page 167 lines 7 to 10**

Menger’s theorem (4.2.17) is a classical theorem in connectivity and plays an important role in network theory. It characterizes the condition for the two dual parameters \(K(x, y)\) and \(A(x, y)\) to share a common value.
The requirement \( xy \notin E(G) \) is not a weakness of the theorem since no x,y-cut exists when x and y are adjacent in G [read the remark after note (ii) here above].

\( (v) \) Page 167 line 15

The suffix G in the symbol \( \kappa_G(x, y) \) indicates that the parameter is with reference to G. This is to avoid confusion since the two parameters \( \kappa(x, y) \) and \( \lambda(x, y) \) are considered with reference to some subgraphs of G later in the proof, in lines 29, 31, and 37 in page 167 and line 3 in page 168 of Textbook DW.

\( (vi) \) Page 167 line 16

The symbol \( N(x) \) stands for the neighbourhood of the vertex x in G. [refer definition 1.3.1 page 34 of Textbook DW].

\( (vii) \) Page 167 line 18

The symbol \( G(V_1) \) stands for subgraph of G spanned by the \( V_1 \subset V(G) \) [refer definition 2.1.1 page 67 of Textbook DW].

\( (viii) \) Page 168 line 6

The symbol \([N(x), N(y)]\) stands for the set of edges having one end in \( N(x) \) and the other in \( N(y) \) [refer line 9 page 152 of Textbook DW].

\( (ix) \) Page 168 lines 8 and 9

Konig-Egervary theorem is a min-max theorem studied in the previous unit. [refer theorem 3.1.16 page 112 of Textbook DW]

Now we see a few applications of Menger’s theorem.

READ DW Page 170 line 1 to page 171 line 25

NOTES:

\( (i) \) Page 170 lines 2 to 3 :

The concept of U-fan helps to obtain a characterization of k-connected graphs [fan lemma by Dirac] using Menger’s theorem.
(ii) **Page 170 lines 4 to 6**

The theorem 4.2.23 [fan lemma], due to Dirac, gives another characterization of k-connected graphs. The proof uses Menger’s theorem.

(iii) **Page 170 lines 22 and 23**

The proof of theorem 4.2.24 uses the Menger’s theorem and the pigeon hole principle. The pigeon hole principle stated in the margin.

(iv) **Page 171 line 10 to page 171 line 24:**

The note gives certain situations in which Menger’s theorem can be applied and also explores the relation between Hall’s theorem and Mengers’s theorem.

Now let us see an example:

**Example 12**: Let us consider the graph given below in Fig. 11 (a)

Here \( n \) (pigeons) = 10 and \( m \) (holes) = 9, so it must be concluded that some hole has more than one pigeon.

Let \( U = (u_1, u_2, u_5, u_9) \). Then an \( x-U \) fan is shown below in Fig. 11 (b)

Another \( x-U \) fan is shown below in Fig. 11 (c)
Now you try the following exercises.

E10) Determine $K(u, v)$ for the following graph.

![Graph](Fig.12)

E11) Let $G$ be a 2-connected graph and let $P$ be a $u - v$ path in $G$. Show that there is a $u - v$ path $Q$ which is internally disjoint with $P$.

E12) Let $v$ be a vertex in a 2-connected graph $G$. Prove that $v$ has a neighbour $u$ such that $G - \{u, v\}$ is connected.

E13) Draw a graph $G$ with connectivity 2 and with 2 vertices $u$ and $v$ connected by 4 internally disjoint paths.

E14) Give an example of a Graph $G$ with connectivity 1 and which contain an $x-U$ fan of size 5.

In the next section we shall discuss network flow.

### 7.7 NETWORK FLOW PROBLEMS

The network systems considered here can be modeled by weighted directed graph only. In a directed graph, the edges are directed in the sense that one vertex of each edge (directed edge) is considered an initial vertex and the other terminal. Directed edges are also called arcs. A weighted graph is one in which capacities (weights) are assigned to every edge. Directed graphs are defined in page 53 weighted graphs in page 95 of the Textbook DW.

In this unit by a network we mean the graph model of a network.

Networks are useful models for several situations such as communication, power and water distribution systems, transportation, etc. In communication systems and transportation systems flow may be allowed in both directions while in power distribution and water distribution normally flow will be allowed in one direction only. Here we confine to the latter type with one source and one sink only. The problem of finding a maximum flow from the source to the sink through the network is discussed.

READ DW Page 176 line 1 to page 180 last line

NOTES:
Network is a weighted connected digraph in which weight corresponds to the capacity of the respective edge. Capacity of an edge is the maximum amount of flow that can be allowed through the edge. Further, two vertices are designated as source and sink respectively. Flow is considered to be from source to sink. Flow at an edge is the actual amount of flow through that edge at any instant. The number $f^+(v)$ is the sum total of the outward flow through all outgoing edges at $v$ and $f^-(v)$ the counterpart. A flow in a network is feasible only if it obeys the capacity constraints namely flow along each edge is non-negative and not exceed its capacity. Conservation constraint assures the equality of total incoming flow and outgoing flow at each vertex other than the source and the sink. It may be noted that these definitions/conditions are intuitive and natural.

$c$ and $f$ are real valued functions on the set of edges and they are respectively called capacity function and flow function. $f^+$ and $f^-$ are real valued functions on the set of vertices of a network.

The value of flow is the excess of total inward flow from the total outward flow at the sink. [Note that some authors don’t allow inward flow to source and outward flow from sink in their definition.]

Example 4.3.3 gives two different feasible flows $f$ and $f'$ in the same network. To obtain the second flow $f'$ from $f$, the flow along the edge $vx$ has been redirected along the edge $vt$ and the flow along the edge $sx$ has been incremented by one. This is possible since $f$ values of $vt$ and $sx$ are less than their capacities.

Recall the definition of augmenting path with reference to a matching studied in unit 6 [line 5 page 109 of textbook DW]. Here also $f$-augmenting path can be defined as a path none of whose edges are saturated by the flow $f$. i.e. it is possible to increase the flow along the path. $E(e)$ is the possible increment in flow along the edge $e$ and the tolerance is the maximum possible increment in flow along the path.

Lemma 4.3.5 page 177 Textbook DW provides a method to revise a flow along a network to obtain a maximum flow.
(vi) **Page 178 lines 1 to 3**

The definition of source/sink cut is at par with the definition of edge cut given in definition 4.1.7 page 152 of **Textbook DW**. Here also S and T form a partition of V(G), but with source in S and sink in T. Definition of capacity [cut capacity] is intuitive.

(vii) **Page 178 line 12**

The definition of net flow out of a subset U of V(G) is also natural. Net flow out of a U is given by the excess of total flow on edges going out of U to the total flow on edges coming into U.

(viii) **Page 178 lines 13 to 16**

Lemma 4.3.7 is important in considering an enhancement in a flow.

(ix) **Page 178 lines 29 and 30**

Corollary 4.3.8 gives theoretical support to an anticipated result.

(x) **Page 179 line 15**

The Ford-Fulkerson labeling algorithm applies lemma 4.3.7 a maximum matching recursively from any given feasible matching in a network.

(xi) **Page 178 line 30**

Example 4.3.10 illustrates the Ford-Fulkerson labeling algorithm using the network given in Example 4.3.3 page 176 textbook DW.

(xii) **Page 180 lines 13 to 15**

Theorem 4.3.11 is the Max-flow Min-cut theorem due to Ford and Fulkerson. This is also a kind of min-max theorem and characterizes maximum flow in terms capacity of a source/sink cut.

This helps verify the maximality of a flow in a network.

Let us look at some examples:

**Example 13:** Let us consider the graph given in the next page (see Fig.13)

![Fig. 13](image_url)

Here C(s x) = 3 and C(s y) = 1.
Example14: Let us look at graph given below

![Graph](image)

\( S = \{s, x, y, z\} \) and \( T = \{p, t\} \)

Then the edges from \( S \) to \( T \) are 2t, yt and 3t. We find that the capacities of each these edges 5.

You can now try this exercise.

E15) Let \( N \) be the network shown below. find a flow on \( N \)

![Network](image)

With this we come to an end to this unit. We now summarises our discussion.

7.8 SUMMARY

In this unit we have covered the following points:

1. The concepts of vertex cuts, edge cuts, disconnecting sets, separating set, blocks, U-fan in a graph are studied.
2. The parameters such as vertex connectivity and edge connectivity of a graph are studied.
3. Classical theorems in connectivity due to Whitney and Menger are discussed.
4. Some applications of Menger’s theorem are discussed.
5. Concept of abstract network, and their characteristics are explained.
6. Certain fundamental theorems related to network and an algorithm to obtain a maximum flow in a network are also explained.
7.9 HINTS/SOLUTIONS

E1) In the given graph \( \{u_1, u_4, u_5\}, \{u_6, u_{10}\}, \{u_8, u_7, u_9\} \) are cuts. The size of a minimum cut is \( z \). Hence the connectivity is \( z \).

E2) We can observe from the following figure.

![Figure 16](image1.png)

That the we can see that removal of a vertex set of size less than three will not disconnect the graph. Also note that \( \{u_1, v_1, u_3\} \) is a vertex cut. Hence connectivity of this wheel is 3. Similarly for any wheel with \( v \) as the central vertex and \( u_1-u_2 \cdots u_n \) as the outer cycle, \( n \geq 4 \), \( \{u_1, v, u_3\} \) is a cut. Hence for any \( n \geq 4 \), the wheel \( W_n \) is of connectivity 3.

E3) One such graph is given below

![Figure 17](image2.png)

E4) Suppose \( \Delta(G) \leq 3 \) and let \( S \) be a minimum vertex cut in \( G \) (\( \delta(S) = K(G) \)). Let \( H_1 \) and \( H_2 \) be two components of \( G - S \). Now \( v \in S \implies d(v) \leq 3 \) and \( v \) has almost neighbour in \( H_1 \) or it has almost one neighbour in \( H_2 \). Then you proceed on the proof for 3-regular graphs \( K(G) = K^1(G) \).

This we have \( \Delta(G) \leq 3 \implies K(G) = K^1(G) \). Thus \( K^1(G) > k(G) \) is possible only if \( \Delta(G) \geq 4 \).

E5) One such graph is given below

![Figure 18](image3.png)

E6) One such graph is given below

![Figure 19](image4.png)
E7) We can give the proof using induction on the number of cut points if there is only one cut point, this is true for, the removal of the cut point decompose graphs into components. Now let us assume this is true when there are \( k \) cut vertices and let \( G \) be a graph with \( k + 1 \) cut vertices. Let \( G_1, G_2, \ldots, G_r \) be components of \( G - u \), where \( u \) is one cut vertex. Then if \( k_1, k_2, \ldots, k_r \) are the number of cut vertices in \( G_1, G_2 \ldots, G_r \), respectively, then \( k_1 + k_2 + \cdots + k_r = k \). By induction hypothesis every components are having fewer cut vertices than blocks. Number of blocks is greater than or equal to \( k_1 + k_2 + \cdots + k_r + r = k + r \) and it can be seen that \( G \) is also having at least \( k + r \) blocks, \( r \geq 2 \). Thus the number of blocks is greater than the number of cut vertices.

E8) i) One such graph is given below

![Graph](image1)

ii) One such graph is given below

![Graph](image2)

iii) One such graph is given below

![Graph](image3)

iv) One such graph is given below

![Graph](image4)

E9) Try yourself. See this is relevant to prove that any cubic graph with a cut vertex is having a cut edge.
10) Try yourself.

11) This follows from Merger theorem. If \( G \) in 2 consider for any \( u, v, \) in \( G \) there are 2 internally disjoint \( u - v \) paths.

12) Let \( G \) be a 2-connected graph and let \( v \in V(G) \). Then \( G - v \) is connected.

Let \( u \in N(v) \) if \( G - \{u, v\} \) is connected, we have done.

If \( G - \{u, v\} \) is not connected Let \( G_1 \) be a component in \( G - \{u, v\} \), and let \( w \in G_1 \) is such that \( w \) is adjacent to \( v \). Let \( x, y \in G - \{v, w\} \) if \( x, y \in G_1 \) and if \( w \) is on an \( x - y \) path then there is on \( x - y \) path in \( G \) internally disjoint with this path. Then if \( v \) is notion this path then there is an \( x - y \) path in \( G - v, w \).

Since \( G^1 = G(v(G_1) \cup \{u, v\}) \) in 2-connected there is path connecting \( v \) and \( w \) in \( G^1 - vw \). Combining this with the fact that \( G - v \) in connected we get that \( G - \{v, w\} \) is connected.

E13) One such graph is given below

![Fig.22](image)

In the above graph there are four internally disjoint \( u.v \) paths but \( G - \{u, v\} \) is disconnected.

E14) One such graph is given below

![Fig.23](image)

Let \( U = \{u_1, u_2, u_3, u_4, u_5\} \), Then there is on \( x - u \) fan of size 5 but \( G - x \) is disconnected, So connectivity of \( G \) is 1.
UNIT 8 COLOURING OF GRAPHS

<table>
<thead>
<tr>
<th>Structure</th>
<th>Page No</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Introduction</td>
<td>53</td>
</tr>
<tr>
<td>Objectives</td>
<td></td>
</tr>
<tr>
<td>8.2 Vertex Colouring</td>
<td>54</td>
</tr>
<tr>
<td>8.3 Structure of Graphs with Large Chromatic Number</td>
<td>60</td>
</tr>
<tr>
<td>8.4 Summary</td>
<td>63</td>
</tr>
<tr>
<td>8.5 Hints/ Solutions</td>
<td>63</td>
</tr>
</tbody>
</table>

**8.1 INTRODUCTION**

Colouring of graphs is among very important topics in graph theory. In fact, people study graph theory to understand and solve problems related to the theory of graph colourings. Beginning with the famous four color conjecture, which is now known to have been solved, there is a large material on graph theory that has practical applications. These include timetable scheduling problem and register allocation.

This unit deals with proper colouring of vertices. A proper vertex coloring is an assignment of colours to the vertices of a graph such that no pair of adjacent vertices are assigned the same colour.

The studies of colouring of edges and even faces of a graph are also done. However, edge colouring is not included in the syllabus. But those who are interested can refer to the seventh chapter in Text book DW. Faces can be defined in case of a special and important class of graphs called planar graphs. We will discuss face coloring in the next unit.

Vertex colouring and a related concept are discussed in Section 8.2. This section also deals with the notion such as chromatic number and chromatic graphs. Loops and multiple edges make no impact on vertex colouring. So in this unit also only simple graphs are considered. In Section 8.3 we discuss the structure of k–chromatic graphs and Mycielski’s construction for producing a sequence of such graphs.

**Objectives**

After Studying this unit, you should be able to

- Illustrate the following concepts
  i) vertex colouring
  ii) k-colouring
  iii) chromatic number;
- State and use Brook’s theorem;
- Apply a greedy colouring algorithm;
- Explain and apply Mycielski’s construction.
8.2 VERTEX COLOURING

Here we shall familiarise you with the concept vertex colouring.

We shall begin with an example.

**Example 1: Storing Chemicals:** A chemical manufacturer wishes to store chemicals in a warehouse. Some chemicals react violently when in contact with each other, and the manufacturer decides to divide the warehouse into a number of areas so as to separate dangerous pairs of chemicals. In the following table, an asterisk indicates those pairs of chemicals that must be kept apart. What is the smallest number of areas needed to store these chemicals safely?

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>b</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>c</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>d</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>e</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>f</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>g</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Let us represent the situation by a graph (see Fig. 1). Vertices are chemicals, edges between two chemicals indicate pairs of chemicals that must be kept apart.

Here the problem is to determine the smallest number of areas needed to store the chemicals safely. We note first that chemicals a, b, c must all be in separate areas, so at least four areas are necessary. In fact, four areas are sufficient, as the graph in Fig. 1 shows; the vertices correspond to the seven chemicals, and two vertices are joined by an edge whenever the corresponding chemicals must be kept separate. If we colour the vertices with the minimum number of colours then four colours are needed, as indicated by the numbers next to the vertices in the giving the graph; the four colours 1, 2, 3, 4 correspond to the four areas.

Thus, we can split the set of chemicals into four disjoint subsets corresponding to the four areas:

\{a, e\}, \{b, f\}, \{c\}, \{d, g\}.

(Other solutions are possible.)

As we already noted in the example above, vertex colouring is an assignment of colours (or labels) to the vertices. For a graph G of order n, one can assign
same colour to all $n$ vertices or $n$ different colours. Both of these are extremes. So colouring with some condition may be of importance. This motivates a colouring such that no pair of vertices adjacent in the graph receives the same colour. Such a colouring is called a proper colouring (proper vertex colouring). If it is possible to find a colouring of the vertices of $G$ using $k$ colours, then $G$ is called $k$-colourable. If $k < n$, then a $k$-colourable graph is $(k + 1)$-colourable also. But you can see that if the edge set $E(G)$ is non-empty then $G$ will not be $1$-colourable. If $G$ contains a triangle as a subgraph then it will not be $2$-colourable, So, for each graph, a minimum number of colours is needed for a proper colouring of its vertices.

This leads to the concept of chromatic number which we shall illustrate now. As we already mentioned above, a graph is said to be $k$-colourable if it is possible to assign colour from a set of $k$ colours to each vertex such that no two adjacent vertices have the same colour. If the graph $G$ is $k$-colourable but not $(k - 1)$-colourable. We say that $G$ is $k$-chromatic graph and that its chromatic number $\chi(G)$ is $k$. So the chromatic number is the minimum number $k - 1$ such that $G$ is $k$-colourable. Hence a graph $G$ is $k$-colourable if and only if $\chi(G) \leq k$. Thus, a $k$-chromatic graph is a graph that need at least $k$-colours whereas a $k$-colourable graph is a graph that does not need more than $k$-colours. You will learn more about them while reading the corresponding portion of Textbook D.W.

### NOTES:

(i) **Page 191 last five lines**

Note that a proper $k$-colouring of a graph $G$ on $n$ vertices always exists for some $k$. Simply give different colours to each vertex which results in a proper $n$-colouring of $G$. In order to show that $G$ is $k$-chromatic which is the same thing as showing that $\chi(G) = k$, we need to prove two things: there is a proper colouring of $G$ in $k$ colours, that is $G$ is $k$-colourable and there is no proper colouring of the vertices of $G$ that uses $k - 1$ or less number of colours.

(ii) **Page 192 Example 5.1.3**

It is easily seen that when we take a fixed vertex $x$ of Petersen graph and consider all the six vertices that are not adjacent to $x$, then the induced subgraph $H$ on these six vertices is 2-regular. Since $G$ contains no cycles whose length is less than 5, we see that $H$ is a 6-cycle $C_6$. We can then use two colours $a$ and $b$ to alternately to colour the vertices of $H$ in two colours (See Fig.2). If we now use the third colour $c$ to colour all the three neighbours of $x$, then $x$ itself can be coloured either $a$ or $b$. This produces a 3-colouring of the Petersen graph and hence it has chromatic number 3.
Graph Theory

The following figure shows a three colouring of the Petersen graph.

![Petersen Graph with Colouring](image)

The labels B, G and R attached to the vertices indicate the colour (blue, green or red respectively) assigned to the vertices.

(iii) **Page 192 first five lines**

It is clear that a graph is two colourable if and only if it is bipartite. Naturally this extends to k-colourable case. This bring a structural property and an optimization problem together.

Note that when we properly colour the vertices of a graph G in k colours, a colour class is formed with the set of all vertices that receive the same colour. We then have k colour classes. Since no two vertices in the same colour class can be adjacent to each other, it follows that the graph induced on each colour class has no edges which is the same thing as saying that each colour class is an independent set.

(iv) **Page 192 Definition 5.1.4**

Any odd cycle $C_{2r+1}$ has chromatic number 3 and deletion of any vertex or any edge from an odd cycle results in a path whose chromatic number is 2. So an odd cycle is 3-critical. Note that this is not true for an even cycle (whose chromatic number is 2).
Since the Petersen graph contains a large number of odd cycles (which have chromatic number 3), cannot be 3-critical (though the Petersen graph is 3-chromatic).

v) **Page 192 and 193 (Definition 5.1.6) and Proposition 5.1.7**

Let $\chi(G) = k$ and assume that we have a k-colouring of the vertices of $G$. If $A_i$ denotes the vertices of the i-th colour class, then $|A_i| \leq \alpha(G)$ for all $i = 1, 2, \ldots, k$.

We thus have

$$n = \sum_{i=1}^{k} |A_i| \leq k\alpha(G)$$

This shows that $k \geq \frac{n}{\alpha(G)}$ and hence $\chi(G) \geq \frac{n}{\alpha(G)}$.

The clique number $w(G)$ gives a lower bound on the chromatic number $\chi(G)$. If $G = K_n$, then the clique number $w(G)$ and $\chi(G)$ are both equal to $n$. However, it is not difficult to construct graphs for which there is not even a complete subgraph with three vertices (and hence $w(G) 2$) but the chromatic number $\chi(G)$ is arbitrarily large.

vi) **Page 193-5.1.9, 5-1-10 and 5.1.11**

Check that the product $Q_{k-1} \square K_2$ actually gives us $Q_k$. Proposition 5.1.11 then shows the chromatic number $\chi(Q_k) = 2$ by making an induction on $k$. One can also see this directly by observing that $Q_k$ is a bipartite.

vi) **Page 194 and 195 5.1.1.12 and 5.1.13**

Greedy algorithm gives an upper bound on $\chi(G)$ by producing a proper vertex colouring of $G$. The trouble with greedy algorithm ‘not being so good’ is the following.

Consider the following path $P_4$ with four vertices $v_1, v_2, v_3$ and $v_4$ shown below. Note that in order to apply the greedy algorithm, the ordering on the vertices is $v_1, v_4, v_3, v_2$.

```
\begin{figure}[h]
\centering
\begin{tikzpicture}
  \node (v1) at (0,0) [circle,fill,inner sep=2pt] {$v_1$};
  \node (v2) at (1,0) [circle,fill,inner sep=2pt] {$v_2$};
  \node (v3) at (2,0) [circle,fill,inner sep=2pt] {$v_3$};
  \node (v4) at (3,0) [circle,fill,inner sep=2pt] {$v_4$};
  \draw (v1) -- (v2) -- (v3) -- (v4);
\end{tikzpicture}
\caption{Fig.4}
\end{figure}
```

It then follows that we give colour 1 to $v_1$, colour 1 to $v_2$ and colour 2 to $v_3$ forcing us to give colour 3 to $v_4$. However, this path being bipartite, has chromatic number equal to 2. The following ordering of the vertices and an application of the greedy algorithm uses only two colours: $g(v_1) = 1, g(v_4) = 2, g(v_3) = 1$ and $g(v_2) = 2$.

Also if we know that $\chi(G) = k$ and hence there is a colouring in which $a_i$ vertices are coloured with colour $i$ for $i = 1, 2, \ldots, k$ with
\[ \sum_{i=1}^{k} a_i = n. \]

Then we can just order the vertices of the graph \( G \) in such a way that the first \( a_1 \) vertices are from the colour class 1, the next \( a_2 \) vertices are from colour class 2 and so on.

Application of the greedy algorithm with this ordering does colour \( G \) with \( k \) colours. The trouble here is how do we know as to which ordering is good (because we do not really know the chromatic number but are trying to find it). In fact, it is possible to recursively construct a very large tree with a very large number of vertices and have a bad ordering on the vertices of the tree so that the number of colours required by the greedy algorithm is very large (and hence tends to infinity). But the chromatic number of any tree is only 2 since a tree is bipartite graph:

\begin{itemize}
  \item [vii)] \textit{Page 195 and 196 Example 5.1.15 and Proportion 5.1.16}
  
  For an interval graph we sequence (order) the intervals by lining them up according to their left end points. Note that the vertices of an interval graph are themselves (or are represented) by open intervals on the real line and are therefore naturally ordered. If two different open intervals have the same left end point, then they are ordered according to their right end points. That is, if \( b < c \), then the interval \( x_1 = (a, b) \) is before \( x_2 = (a, c) \). If at some stage of running of the greedy algorithm (with the ordering we just described), a vertex \( x = (a, b) \) gets colour \( k \), the chromatic number of \( G \), then it must have neighbours \( x_i = (a_i, b_i) \) that have already been scanned that \( x_i \) gets colour \( i \) where \( i = 1, 2, \ldots, k - 1 \). This forces \( a_i \leq a < b_i \). It then follows that for \( i \neq j \) the open interval \( (a_i, b_i) \) and \( (a_j, b_j) \) must intersect in a non-empty set. Hence \( x \) along with \( x_1, x_2, \ldots, x_{k-1} \) forms a clique of size \( k \) and therefore \( w(G) \geq k \). Since we always have \( \chi(G) \geq w(G) \), we get \( \chi(G) = w(G) \). Our arguments also show that at no stage a vertex can receive colour \( k + 1 \) for, in that case, we must have a \( (k + 1) \)-clique containing \( x \) forcing \( \chi(G) \geq w(G) \geq k + 1 \). This shows both things: the greedy algorithm uses only \( \chi(G) \) colours as also \( \chi(G) = w(G) \).

  \item [viii)] \textit{Page 196 line 16 upto line 20}
  
  A 3-critical chromatic subgraph of the 3-chromatic Petersen graph is a cycle \( C_5 \). There is a large number of such 5-cycles in the Petersen graph.

  \item [ix)] \textit{Page 197 Theorem 5.1.22}
  
  Note that the exceptions stated in Brook’s theorem are essential. The complete graph \( G = K_n \) has chromatic number \( n = 1 + \Delta(k_n) \) and the same is true for an odd cycle whose chromatic number is 3.
We shall make a remark now.

**Remark:** There is a simple method for obtaining a lower bound for $\chi(G)$. We can look for the largest complete subgraph in $G$, say $K_n$, for some $n$. Then $\chi(G) \leq n$.

Let us see an example.

**Example 2:** Let us show that the following graph $G$ is 4-colourable and establish a 4-colouring of $G$. What can we say about $\chi(G)$?

![Fig. 5 (a)](image)

**Solution:** We observe that the above graph contains the complete graph $K_4$. Therefore $\chi(G) \geq 4$. Now we apply Brook’s Theorem. Note that $G$ is a connected simple graph with maximum vertex-degree, $d = 4$. Therefore by Brook's Theorem $\chi(G) \leq 4$. Thus $\chi(G) = 4$. A 4-colouring is illustrated in the following theorem.

![Fig. 5 (b)](image)

Try these exercises now.

---

**E1)** Find $\chi(G)$ for the following:

i) the complete graph $K_7$

ii) the 4-cube $Q_4$

iii) the cycle graph $C_9$

**E2)** Which of the following statement are true or false? Give justification for your answer.

i) If $\chi(G) = n$ for some $n$, then $G$ contain the complete graph $K_n$ as a subgraph.

ii) If $G$ contain the complete graph $K_n$ as a subgraph, then $\chi(G) \geq n$. 

---

59

Colouring of Graphs
8.3 STRUCTURE OF GRAPHS WITH LARGE CHROMATIC NUMBER

In this section we shall discuss about the structure of graphs without triangles that have arbitrarily large chromatic number. If a connected graph $G$ does not have a triangle as a subgraph, the lower bound $w(G)$ is only 2 and the chromatic number could be large, in which case the lower bound $\chi(G) \geq w(G)$, is not every helpful. In fact starting from a triangle-free $k$-chromatic graph $G$, it is always possible to obtain a triangle-free $(k+1)$-chromatic graph, for any $k$. This is illustrated in Mycielski construction method which we shall explain in this section.

We shall first consider an example to illustrate the method.

**Example 3:** We start from the triangle-free $K_2$ consisting of vertices $v_1$ and $v_2$ See Fig. 6 (a). The new nonadjacent vertices are $u_1$ and $u_2$ along with $w$, which is adjacent of both of them See Fig. 6 (b). We join $u_1$ to $v_2$ and $u_2$ to $v_1$. At the end of the first iteration, we get a cycle with five vertices that is a 3-chromatic triangle-free graph See Fig. 6 (c). For the next iteration, we begin with a cycle with vertices $v_i$ ($i = 1, 2, 3, 4, 5$) and introduce an independent set of five new vertices $u_i$ ($i = 1, 2, 3, 4, 5$) see Fig. 6 (d) and new vertex $w$ that is adjacent to each of these new vertices. The resulting triangle-free 4-chromatic graph (known as the Grotsch graph) Obtained at the end of the second iteration is shown in Fig. 6 (d).
NOTES:

(i) Page 205 line 5, Definitions 5.2.1: and line 12, Theorem 5.2.3

Mycielski graph $G_k$ has chromatic number $k$, the construction illustrated in 5.2.1 is iterative and the proof of theorem 5.2.3 is inductive and by contradiction.

(ii) Page 206 line 11, Remark 5.2.4

Clearly, a graph $G$ (on $n$ vertices) has chromatic number 1 if and only if $G$ has no edges. Suppose $G$ has edges. Then the chromatic number is at least 2. If the chromatic number is equal to 2, then we have two colour classes say $X$ and $Y$ and both $X$ and $Y$ are independent sets showing that a graph with chromatic number 2 is indeed bipartite. Conversely if $G$ is bipartite with bipartition $(X, Y)$ then we can colour all the vertices in $X$ in colour 1 and all the vertices in $Y$ in another colour showing that $\chi(G) = 2$. We thus have; $G$ has chromatic number 2 if and only if it is bipartite. In turn, recall that $G$ is bipartite if and only if $G$ has no odd cycle. Hence showing that $\chi(G)$ equal 2 is equivalent to showing that $G$ is bipartite. Checking if $G$ is bipartite is not very difficult. Assume that $G$ is connected; if this is not the case then we can look at the connected components of $G$. The breadth first search tree rooted at a vertex $x_0$ gives us the distance of from $x_0$. One can then partition the vertex set into two parts $X$ and $Y$ as follows. $X$ consists of those vertices that are at an even distance from $x_0$ (this also includes $x_0$). If there is no edge inside $X$ or inside $Y$, then $G$ is a bipartite graph with bipartition $(X<Y)$. To sum up, checking when $\chi(G) = 2$ has an easy answer. No similar procedure exists to for $G$ to have a chromatic number $k$ where $k \geq 3$.

Thus you might have understood that, by starting with 2-chromatic graph $K_2$. The Mycielski’s construction provider, for all $k \geq 2$, a triangle-free $k$-chromatic graph $G'$ on $3.2^{k-2} - 1$ vertices.

We shall now look at an example..

**Example 4:** Let us show that if $G$ is colour-critical, then $G'$ is also colour-critical.
Solution: Let $G$ be a $k$-color critical graph and $G'$ be the graph obtained from $G$ by Mycielski’s construction. Then $\chi(G') = k + 1$ by the construction. We have to show that $G'$ is $k+1$-colour critical. i.e. $\chi(G) < \chi(G - e)$ for every edge $e$ in $G'$. For, by the construction procedure procedure $V(G') = \{v_1, v_2, v_3, \ldots, v_n, u_1, u_2, u_3, \ldots, u_n, w\}$ and an edge in $G'$ is of the form $v_i, v_j \in E(G)$, $v_i u_j$ or $u_i w$.

Case 1: $e = v_i v_j$ in this case $e$ is an edge in $G$ and $G - e$ is $k-1$ colourable since $G$ is $k$-critical. Since $G$ is $k$-critical, there is a $k$-colouring of $G$ in which $v_j$ receive a unique colour (say $c_j$) as given in 5.2.12 (a). Let $c_i$ be the colour received by $v_i$. So the colour $c_i$ is not assigned to any vertex adjacent to $u_j$ in $G' - e$.

Now from the $k+1$ colouring of $G'$, obtain a $k$ – colouring of $G' - e$. by recolouring $u_j$ using the colour $c_i$ and $w$ using the colour $c_j$.

Case 2: $e = v_i u_j$
Consider a $k$-colouring of $G$ with $v_i$ assigned a colour (say $c_j$) that was not assigned to any other vertex, as given in 5.2.12 (a)

In $G'$ the vertex $u_j$ also has the colour $c_j$ and appears at no other $u_i$.
In $G' - e$ , assign the colour of $v_i$ to $u_j$ and the colour $c_j$ to $w$.
This result in a $k$-colouring of $G' - e$.

Case 3: $e = u_j w$
Consider a $k$-colouring of $G$ with $v_i$ assigned a colour (say $c_j$) that was not assigned to any other vertex, as given in 5.2.12 (a)
So in $G'$, $u_j$ also receives the colour $c_j$. Re-colour $w$ in $G' - e$. by $c_i$ , which is possible since $w$ is not adjacent to $u_j$ in $G' - e$.

In each case we produced a $k$-colouring of $G' - e$ and the edges chosen is arbitrary.

This is proves the result.

Now we want you to try an exercise.

---

E4) Let $G_3, G_4, \ldots$ be the graph obtained from $G_2 = K_2$, using Mycielski’s construction. Show that each $G_k$ is $k$-critical.

With this we come to an end of this unit. We shall now summarise the points discussed in this unit.
8.4 SUMMARY

In this unit, we have covered the following points:

1. We have explained the concepts of k-colouring, chromatic number \( \chi(G) \) and clique number \( w(G) \), for a graph \( G \).

2. A greedy algorithm for finding an upper bound for \( \chi(G) \) for a graph \( G \).

3. Brook’s theorem without proof for obtaining an upper bound for \( \chi(G) \) for a graph \( G \).

4. Mycielski’s construction which provides a triangle-free \((k+1)\)-chromatic graph for any \( k \).

8.5 HINTS/SOLUTIONS

E1) i) \( \chi(K_7) = 7 \)
ii) \( \chi(Q_4) = 2 \)
iii) \( \chi(C_9) = 3 \)

E2) i) False. For example, consider the cyclic graph \( C_5 \). Then \( \chi(C_5) = 3 \), but it contains no triangle.

ii) True. This is because \( G \) contains \( K_n \) as a subgraph.

E3) **Hint:** Note that \( K_2 \) is two colour critical. So by Example 4 (and also by observation), \( G_3 \) is colour critical. Moreover by Example 4, \( G_{k+1} \) is \( k+1 \)-critical whenever \( G_k \) is \( k \)-critical. The result follows by mathematical induction.
UNIT 9 PLANAR GRAPHS

Structure                          Page No

9.1 Introduction                  65
   Objectives                     
9.2 Embeddings and Planarity      66
9.3 Dual Graphs                   68
9.4 Euler’s Formula               70
9.5 Colouring of Planar Graphs    73
9.6 Summary                       75
9.7 Hints/Solutions               75
9.8 Appendix-Four Colour Problem  78

9.1 INTRODUCTION

The study of graph layouts and embedding of graphs on surfaces is termed as topological graph theory. Topology Graph Theory was first described by Euler in 1736 and later developed by Kuratowski, Whitney and others.

In earlier days the study of the concept of planarity was motivated by famous problems such as four colour problem and the three utility problems, etc. Nowadays this concept is also applied in the design of circuit layouts on silicon chips!

A graph whose diagram can be drawn on a surface, on the plane or on the surface of a sphere, without over-crossing of edges is called a planar graph and such a drawing is called planar representation of the graph. In this unit we will investigate characteristics of planar graphs. We will study concept of dual graphs and colouring of planar graphs. Recall that the colouring of graphs is discussed in the previous unit, i.e. Unit 8. We shall also discuss here the famous four colour problem and will give a proof of the 5 colour theorem. A brief history of the four colour problem is also given as an Appendix.

Objectives

After studying this unit, you should be able to

- identify planar graphs and non planar graphs;
- prove that $K_5$ and $K_{3,3}$ are not planar;
- identify the faces of a plane graph;
- draw the dual of a plane graph;
- apply Euler’s formula for plane graphs;
- explain four colour problem for planar graphs.
9.2 EMBEDDINGS AND PLANARITY

In this section, we study planar embeddings of planar graphs and two important graphs which play key roles in the characterization of planar graphs.

You can start reading the Text Book DW.

READ DW page 233 line 1 to page 235 line 21

NOTES:

(i) Page 233 line 1

Topological graph theory deals with embeddings of graphs on surfaces which resulted in the theory of planar graphs.

(ii) Page 233 line 2

The four colour problem was about the minimum number of colours necessary for the coloring of the regions in a map with no pair of region sharing boundary receives the same colour.

(iii) Page 233 line 8

The gas-water-electricity problem discussed in Example 6.1.1 of Textbook DW is called the three utility problems. The question is whether it is possible to lay pipes/cables from the source of each of the three utilities to three different consumers without over-crossing the lines. The answer is affirmative since its graph model is $K_{3,3}$, which cannot be drawn on a plain without over-crossings. You may try various drawings and convince the fact yourself intuitively. You will see a formal proof later in proposition 6.1.2 page 234 of Textbook DW. A drawing of $K_{3,3}$ is given in page 233 and another on page 234 of Textbook DW.

(iv) Page 234 line 5

$K_5$ and $K_{3,3}$ are the two graphs responsible for non-planarity of graphs. They and their subdivisions forbid a graph to be planar. A geometrical proof of non-planarity of $K_5$ and $K_{3,3}$ is given in 6.1.2.

First draw a spanning cycle (cycle containing all the vertices, which exists both the cases) in the graph then try to draw the remaining edges. It can be seen that some of the remaining edges cannot be drawn without over-crossing. An analytical proof is given in Example 6.1.24 as a consequence of Theorem 6.1.23 of Textbook DW.
A planar graph can be drawn on the plane without over-crossing of edges. Over-crossing means the edges have common point in the plane other than their end vertices. Such a drawing is called planar embedding (some authors call plane embedding) of the graph.

A planar graph may have more than one planar embeddings.

For example, Fig. 1 given below contains three drawings of the planar graph $K_4$, but only the second and the third are planar graphs.

![Fig. 1](image)

Concepts of open set, region and faces are intuitive. Open set is similar to that in the real plane. Region is a connected open set and face is interior of cycle (or possibly closed walk in case of cut edge) not containing vertices or edges not belonging to the cycle. A cycle has two faces, inner (bounded) and outer (unbounded) and a tree has only one face. Fig. 2

![Fig. 2](image)

Consider the graph in Fig. 2. It is a planar embedding of a graph with four vertices $u, v, w, x$ and four edges $a, b, c, d$. There are two faces for this graph viz. the inner and outer. The cycle uvw is the boundary of the inner face and the boundary of the outer face is the walk $ubwdxwcva$, which is not a cycle. If one draws the vertex $x$ inside the cycle $uvw$, then the edge $d$ and the vertex $x$ will be on the boundary of the inner face. Whenever we count the faces of a planar embedding, the outer surface also need to be counted.

Now we want you to try some exercises.

E1) Show that every subgraph of a planar graph is planar.

E2) Is every subgraph of a non-planar graph non-planar? Justify.
E3) Show that the graph given below is planar.

![Graph](image)

E4) Determine all values of $n$ such that the complete graph $K_n$ is planar.

In the next section we shall consider the ‘dual’ of a graph.

### 9.3 DUAL GRAPHS

Duality is an important and interesting concept in the study of planar graphs. Let us try to understand this concept. A plane graph $G$ partitions the rest of the plane into a number of connected regions; the closures of these regions are called the faces of $G$. Fig. 4 shows a planar graph with six faces, $f_1, f_2, f_3, f_4, f_5,$ and $f_6$. The notion of a face applies also to embeddings of graphs on other surfaces. We shall denote by $F(G)$ and $\phi(G)$, respectively, the set of faces and the number of faces of a plane graph $G$.

![Fig. 4](image)

Each plane graph has exactly one unbounded face, called the exterior face; in the plane graph of Fig. 4, $f$ is the unbounded face (i.e. the face which is not shaded).

Let us try to understand the concept of duality through an example. We shall consider the planar embedding $G$ of the cube as given in Fig. 5 (a).

![Fig. 5](image)
Now we place a new vertex within each face, (including the unbounded face) and join the pairs of new vertices in adjacent faces. Then we obtain the graph G*, which is the planar embedding of an octahedron (see Fig. 5. (c). Then G* is called the dual of G. The new vertices are represented by small circles, and lines joining them are indicated by dash lines. [See Fig.5 (b) and 5 (c)].

To understand the formal definition, you can start reading Textbook D.W.

**NOTES:**

i) **Page 236 Line 16 to page 237 line 26**

A planar embedding of \( K_4 \) and its dual are given by bold and dashed edges respectively in Fig.6. Note that the dual of \( K_4 \) is itself.

![Fig. 6](image)

The dual of a planar graph depends on the planar embedding and hence need not be unique.

In a dual, leaves will appear as loops and vertices of degree two as multiple edges.

The dual of the graph in Fig. 2 is the graph with two vertices, triple edge joining them and a loop at one of the vertices. The dual of a tree with \( e \) edges is just a vertex and \( e \) loops at it. The dual of a cycle of length \( n \) is simply two vertices and \( n \) edges joining them.

ii) **Page 237 Line 27 to page 238 line 22**

Theorem 6.1.12 in Textbook DW states that irrespective of the drawing, the sum of lengths of faces of a planar embedding is twice the number of edges. The counting method discussed above is justified by this result.

While counting the edge on the boundary of a face, cut edges are to be counted twice as both sides of it represent same face, probably outer. If there is a leaf in a planar graph, its planar embedding can be drawn with
either as the leaf is on the boundary of the outer face or on the boundary of an inner face.

iii) **Page 239 Line 9 to page 239 line 24**

Theorem 6.1.16 of *Textbook DW* gives two characterisations for the dual of a planar graph to be Eulerian.

iv) **Page 239 Line 25 to page 240 last line**

A particular class of planar graphs, outer planar graphs, is discussed here. They are planar graphs whose planar embeddings can be drawn in such a way that all vertices lie on the boundary of the outer face.

Read the proposition 6.1.18 in *Textbook DW* as “the boundary of the outer face of a 2-connected outer plane graph is a spanning cycle”.

The proposition 6.1.19 in *Textbook DW* gives two planar graphs which are not outer planar and the proposition 6.1.20 gives a necessary condition for outer planarity, which is helpful in checking outer planarity of a simple graph.

Here are some exercises for you.

E5) The following diagrams show two different plane drawing of the same planar graph. Show that their duals are not isomorphic.

![Fig.7](image)

E6) Which of the following statements are true? Justify your answer.

i) If a planar graph has a cut vertex, then its dual has a cut vertex.

ii) If the dual of the planar graph has a cut vertex, then the graph itself has a cut vertex.

In the next section we shall consider a remarkable simple formula that relates the number of edges, vertices and faces of a planar graph.

### 9.4 Euler’s Formula

Here we are going to discuss the formal proof of a formula that has been taught in school classes associated with the study of solids. It is the formula relating the vertices \(n\), edges \(e\) and faces \(f\) in planar graph and
the formula is \( n - e + f = 2 \). It is easy to see that a stereographic projection of a solid on plane gives a planar graph with vertices of the solid as vertices of the graph and edges of the solid as edges. To learn more about this, you can start reading the Textbook DW.

**NOTES:**

i)  **Page 241 Line 1 line 16**

   We give an alternative proof of Theorem 6.1.21 on page 241 of Textbook D.W.

   **Proof:** we use induction on \( e \), the number of edges.  

   **Basic Step:** \( e = 0 \)

   \( G \) is connected and so \( G = K_1 \)

   \[ \therefore \ n = 1 \text{, and } f = 1 \]

   \[ \therefore \ n - e + f = 2 \]

   **Induction step** (\( e > 1 \)): Now let \( G \) be a graph as given in theorem and assume that the theorem is true for all connected plane graphs with less than \( e \) edges.

   **Case 1:** If \( G \) is a tree, then \( n = e + 1 \) and \( f = 1 \)

   So \( n - e + f = e + 1 - e + 1 = 2 \)

   **Case 2:** If \( G \) is not a tree, then \( G \) contains a cycle. Let \( x \) be an edge of some cycle in \( G \) and consider \( G' = G - x \). Since \( x \) being an edge on a cycle, it is in the boundary of two faces of \( G \) and hence its deletion merges the two faces. So \( G' \) has \( n \) vertices, \( e - 1 \) edges and \( f - 1 \) faces only. Therefore by induction hypothesis, \( G' \) satisfies the Euler’s formula i.e. \( n - (e - 1) + (f - 1) = 2 \) and consequently \( n - e + f = 2 \). Hence the theorem is proved.

   □□□

ii)  **Page 241 Line 17 to page 242 line 24**

   It has been noted that the dual of planar graph depends on the planar representation. But the total number of vertices, edges and faces in the dual remain constant irrespective of the planar representation.

   The Euler’s formula given in 6.1.21 is only for connected graphs.
For a disconnected graph the components share the outer face and hence the contribution by the number of components is also to be incorporated. So for a disconnected graph with \( k \) components the Euler’s formula generalizes as \( n - e + v = k + 1 \). In case of connected graph \( \equiv 1 \) and it reduces to \( n - e + v = 2 \).

Here we give proof of the generalized formula which can be stated as follow.

**Theorem 1** Let \( G \) be a plane graph with \( n \) vertices, \( e \) edges, \( f \) faces and \( k \) components. Then \( n - e + f = k + 1 \).

**Proof:** Consider a plane embedding of \( G \) such that the exterior face is shared by all components. Let the components be \( G_1, G_2, \ldots, G_k \).

Let \( G_i \) have \( n_i \) vertices \( e_i \) edges and \( f_i \) faces for \( i = 1, 2, \ldots, k \).

Then for each \( i = 1 \) to \( k \) we have \( n_i - e_i + f_i = 2 \) by Euler’s formula. Summing over \( i \), \( \sum_{i=1}^{k} n_i - \sum_{i=1}^{k} e_i + \sum_{i=1}^{k} f_i = 2k \) \( \ldots \) \( \ldots \) \( \ldots \) (1)

But, \( \sum_{i=1}^{k} n_i = n \), \( \sum_{i=1}^{k} e_i = e \) and

\( \sum_{i=1}^{k} f_i = f + (k - 1) \) since the exterior face is counted once for each of the \( k \) components in \( \sum_{i=1}^{k} f_i \) and it counts only once in \( f \).

So (1) becomes \( n - e + (f + k - 1) = 2k \)

Thus \( n - e + f = k + 1 \)

\( \square \)

**iii) Page 241 Line 27 to Line 28**

Theorem 6.1.23 gives necessary conditions that can be easily checked. The first is for general graphs while second condition is much stronger only for triangle free graphs. A triangle is a \( K_3 \) subgraph.

**iv) Page 242 Line 3 to Line 5**

Example 6.1.24 gives an analytical proof of non-planarity of \( K_5 \) and \( K_{3,3} \) using the necessary conditions described in 6.1.23.

We shall now use Euler’s theory to provide another interesting result.
**Proposition 1:** If $G$ is a connected planar simple graph with at least 3 vertices and $e$ edges whose shortest cycle length is 5, then $e \leq \frac{5}{3}(n - 2)$.

**Proof:** We first note that the shortest cycle length in $G$ is 5. Therefore the degree of each face in a plane drawing is at least 5, so that $2e \geq 5f$ or $f \leq \frac{2}{5}e$.

Combining this with Euler’s formula, $f = e - n + 2$ we get,

\[
\frac{2}{5}e \leq n - 2
\]

\[
\frac{3}{2}e \leq n - 2
\]

\[
e \leq \frac{5}{3}(n - 2).
\]

Hence the result.

Here are some exercises for you.

---

**E7)** Let us verify Euler’s formula for the following graphs

![Fig 8](image)

(a) ➔ (b)

---

**E8)** Let $G$ be a plane graph with $n$ vertices, $e$ edges and $f$ faces. If every face of $G$ is a $k$-cycle, then $e = \frac{k(n - 2)}{k - 2}$.

**E9)** Use proposition 1 to show that the Peterson graph is not planar.

---

In the next section, we discuss colouring of planar graphs.

### 9.5 COLORING OF PLANAR GRAPHS

Recall the concept of vertex coloring of graphs studied in Unit 8.

Coloring of planar graphs is related to coloring of a maps. A proper coloring of a map means an assignment of colors to the regions in the map such that no pair of regions sharing boundaries receive the same color. For every map it is possible to obtain a planar graph whose vertices representing the regions and two vertices are joined by an edge if and only if the respective region share...
boundary. Then the map coloring problem becomes the vertex coloring problem in planar graphs.

There was a classical problem called the Four Colour Problem paused in 1852 by Francis Guthrie, a student of geography, about the minimum number of colours necessary for a proper colouring of a map. The problem was communicated to De Morgan through his brother Frederic Guthrie. Cayley published a paper on this problem in 1879 and outlined the difficulties that lie in obtaining a solution. In the same year A.B. Kempe came forward with a proof. After ten years, in 1890, P.J. Heawood detected an error in the proof and he modified it and proved the sufficiency of five colours – the five colour theorem. Finally, after about a century, in 1979 Appel and Haken put an end by proving the sufficiency of four colours and hence now we have the four colour theorem.

You will learn more about this while reading the Textbook D.W.

**READ DW Chapter, Section, from Page 257 - 258**

**NOTES:**

1) **Page 257 Line 7**

Here we shall prove the following theorem.

**Theorem:2** Every simple connected planar graph with n vertices has at most $3n - 6$ edges. (Hence such a graph has a vertex of degree at most 5).

**Proof:** $e \leq 3n - 6$ (by Theorem 6.1.23)

$\therefore 2e \leq 6n - 12$

$v \sum\limits_v \deg v \leq 6n - 12$ where $\deg v$ is the degree of any vertex $v$.

Since $G$ is connected, $\deg v \geq 1 \ \forall \ vertices \ v$. If all the vertices have $\deg \geq 6$, then $\sum\limits_v \deg v \geq 6n$.

But $\sum\limits_v \deg v \leq 6n - 12$

Hence there exists a vertex of degree at most 5. In fact we can say that at least three vertices have degree less than 6.
Here are some exercises for you to try.

**E10)** Which of the following statements are true? Justify your answer
i) The chromatic number of a planar graph cannot exceed 4.
ii) Every planar graph is not 6-colourable.

**E11)** Check whether the Four-Colour problem holds for the following graph.

![Fig.9](image)

### 9.6 SUMMARY

In this unit, we have covered the following points:

1. The concepts of planarity, planar embedding, outer planarity, faces, dual of a plane graph.
2. Non planarity of $K_5$ and $K_{3,3}$
3. If $\ell(F_i)$ denotes the length of face $F_i$ in a plane graph $G$, then
   
   $2e(G) = \sum \ell(F_i)$

4. Edges in a plane graph $G$ form a cycle in $G$ if and only if the corresponding dual edges form a bond in $G^*$.

5. $K_4$ and $K_{2,3}$ are planar but not outer planar.

6. The Euler’s formula.

7. Necessary relationship between the order and size of a planar graph in the general situation and in triangle free graphs.

8. Every planar graph is 5-colourable.

### 9.7 HINTS/SOLUTIONS

**E1)** Let $G$ be a planar graph. Let $H$ be a subgraph of $G$. Then $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. If $G$ can be drawn on a plane without crossings then $H$ also can be drawn on a plane without crossings. Hence $H$ is planar.

**E2)** $K_5$ is non-planar but $K_5 \setminus \{e\}$ is planar where $e$ is any edge of $K_5$. 
E3) The given graph can be drawn as given below two edges meet except at a vertex with which they are both incident.

E4) Since $K_4$ is planar and $K_n$ is a subgraph of $K_4$ when $n \leq 4$, $K_n$ is planar when $n \leq 4$. Since $K_5$ is non-planar, and $K_5$ is a subgraph of $K_n$ when $n \leq 5$, $K_n$ is non-planar when $n \leq 5$.

E5) The dual graphs are given respectively in Fig. 10 (a) and (b).

![Fig.10](image)

Their degree sequences are $(3,3,3,3,5)$ and $(3,3,3,4,4)$ and therefore they are not isomorphic.

E6) i) If a plane graph $G$ has a cut vertex then if its dual has a cut vertex.

This statement is not true. The following figure illustrates this.

![Fig.11](image)

In Fig.11 $G$ has a cut vertex $v$. But its dual $G^*$ has no cut vertex.

ii) This is not true as the following example shows.

![Fig.12](image)

In Fig.12 $G^*$ has a cut vertex $v$. But $G$ does not have a cut vertex.

E7) i) Here $n= 10$, $e = 15$ and $f = 7$.

ii) Hint: Direct Verification.
E8) Every face of $G$ is a $k$-cycle. Hence every edge lies on the boundary of exactly two faces.

\[ \therefore 2e = f \cdot k \]
\[ \therefore f = \frac{2e}{k} \]

By Euler’s formula, $n - e + f = 2$

i.e. $n - e + \frac{2e}{k} = 2$

i.e. $e \left(\frac{2}{k} - 1\right) = 2 - n$

\[ \therefore e = \frac{k(2 - n)}{2 - k} \]

i.e. $e = \frac{k(n - 2)}{k - 2}$

E9) Suppose that the Peterson graph is planar. Then the inequality in example 2 becomes

$15 \leq 13 \frac{1}{3}$, which is not possible. Hence the peterson graph is not planar.

E10) i) True – since any planar graph is plane graph.

ii) False – follows from (i) or it can be shown directly that every planar graph is $G$-colourable.

E11) We first label the vertices $a, \ldots, f$ as follows:

![Fig.13](image.png)

Then we successively colour the vertices as

- Vertex $a$ with colour green
- Vertex $f$ with colour blue
- Vertex $c$ with colour green
- Vertex $d$ with colour blue
- Vertex $e$ with colour pink
- Vertex $f$ with colour yellow

![Fig.14](image.png)

All the vertices are now coloured. We thus obtain the 4-colouring of $G$ as shown in Fig.14.
9.8 APPENDIX – A BRIEF HISTORY OF THE FOUR COLOUR PROBLEM

The subject was first raised in 1852 by a part-time mathematician, Francis Guthrie, who was colouring in a map of the counties of Britain. He was intrigued by the fact that four colours appeared to be sufficient regardless of the complexity of boundary shapes or how many regions had a common border. He passed the problem on to University College London to the eminent De Morgan, who in turn passed it to the great William Hamilton. Hamilton was unable to invent a map which required five colours, but neither could he prove that no such map existed.

Like Fermat's Last Theorem this apparently trivial problem generated a great deal of interest and activity. In 1879 a British mathematician, Alfred Kempe, published a 'proof' that was accepted by the mathematics establishment until in 1890 Percy Heawood of Durham University showed that the so-called proof was fundamentally flawed. The search continued, and like Fermat's theorem led to great advances in number theory, so the four-colour problem gave a stimulus to the new and increasingly important topic of topology.

The first breakthrough in the four-colour problem came in 1922, when Philip Franklin ignored the General problem and settled for a proof which showed that any map containing 25 or fewer regions required only four colours. This was extended in 1926 to 27 regions, in 1940 to 35 regions, and in 1970 to 39 regions. Then in 1976 two mathematicians at the University of Illinois, Haken and Appel, came up with a new technique which would revolutionise the concept of mathematical proof.

Haken and Appel used the ideas of Heinrich Heesch that the infinity of infinitely variable maps could be constructed from a finite number of finite maps. They reasoned that by studying these building-block maps it would be possible to attack the general problem.

This proved very difficult in practice to achieve, because the number of building block configurations could not be reduced below 1482. To crank through all the permutations that might occur with this number of configurations would take a lifetime. Enter the age of the computer. In 1975, after five years of working on the problem, they turned to the new number-cruncher and in 1976 after 1200 hours of computer time they were able to announce that all 1482 maps had been analysed and none of them required more than four colours.

AND THUS WE HAVE THE LONG AWAITED FOUR COLOUR THEOREM.

The problem with this type of proof is that only another computer can carry out the customary check on its validity. Some mathematicians are most reluctant to accept it because no one, however patient, could work through the exposition line by line and verify that it is correct. It has been disparagingly referred to as a 'silicon proof'. The fact is, however, that mathematicians will increasingly have to rely on such methods. The age of the purist of pure logic as the only acceptable technique in mathematics has probably passed.
UNIT 10 HAMILTONIAN GRAPHS

10.1 INTRODUCTION

Consider the problem of organising a tour to visit different places. If possible it may be better to chart out a roundtrip so as to visit every place exactly once, going from one place to another without passing through a place that has already been visited. The question is whether such a journey is possible or not.

To model this system consider a graph with vertices correspond to places and an edges joining two vertices if the respective places are accessible from each other without passing through any other place of visit. The problem has a solution if the graph contains a spanning cycle. A spanning cycle in a graph is called a Hamiltonian cycle and a graph having a Hamiltonian cycle is called a Hamiltonian graph.

This concept is similar to that of Eulerian graph where the graph contains an Eulerian circuit.

From the definition of a line graph of a graph [Definition 4.2.18, page 168 Textbook DW], it appears intuitively that an Eulerian circuit in a graph G is a Hamiltonian cycle in the line graph of G. Unfortunately, there is no general easy way to check whether a graph is Hamiltonian or not. But there are many necessary conditions for a graph to be Hamiltonian. Sufficient conditions are also there to check whether a graph is Hamiltonian or not. Some of these conditions are discussed in this unit. But a condition which is both necessary and sufficient is yet to be solved. It is an open problem for several years.

Loops and multiple edges are irrelevant in the discussion and disconnected graphs forbid spanning cycles. So in this unit we consider connected simple graphs only.

Objectives

After studying this unit, you should be able to

- find Hamiltonian cycles in graphs of small order;
- state and apply necessary conditions for the existence of a Hamiltonian circuits;
- state and apply sufficient conditions for the existence of a Hamiltonian circuit;
10.2 NECESSARY CONDITIONS

As stated earlier there is a bunch of conditions that can be checked for a graph to be Hamiltonian or not. A few necessary conditions are discussed here.

Before we discuss these conditions, let us see some examples.

**Example 1:** In any complete graph with order greater than 2, it is easy to find a spanning cycle. Start with any vertex, trace through a neighbouring vertex that has not been passed previously, till all the vertices are covered and then pass on to the beginning vertex.

**Example 2:** For every \( n \), the cycle graph \( C_n \) (a cycle graph is a graph with single cycles) is Hamiltonian (See Fig 1 below).

**Example 3:** The dodecahedron given in Fig.2 is Hamiltonian.

Why don’t you try it by yourself? In fact this problem was turned into a game by one of the famous Mathematician of 1915 century Sir William Rowan Hamilton (1805-1865) with whom the term Hamiltonian cycles are attached. Hamiltonian, who described, in a letter to his friend Graves, a mathematical game on the dodecahedron (See Fig 2) in which one person sticks five pins in any five consecutive vertices and the other is required to complete the path so formed to a spanning cycle.

In the next example we will give an example of a graph which is not Hamiltonian.

**Example 4:** Consider the graph given on the next page. This graph is called Herschel graph. This graph is not Hamiltonian. You might have noticed that it is bipartite and has an odd number of vertices. Therefore it is not Hamiltonian.
You will learn more about Hamiltonian graph while reading the Textbook DW.

READ TEXT BOOK DW Chapter 7, Section 7.2 pages 286 line 13 to page 288 line 16

NOTES:

(i) Page 287 line 1

The concept of 2-connectedness has been studied by you in Unit 7 [refer page 161 of Textbook DW]. A characterization of 2-connected graphs is given in Theorem 4.2.2 of Textbook DW. It reads ‘A graph G having at least three vertices is 2-connected if and only if each pair of vertices \( u, v \in V(G) \) there exist internally disjoint \( u,v \)-paths in G.

In a Hamiltonian graph, there is a cycle containing all vertices, vertices. This provides two paths, one clockwise and other anticlockwise, joining any pair of vertices. Thus, every Hamiltonian graph is 2-connected or in other words, a block.

Even though this is not numbered as a result in the Textbook DW, it is a necessary condition, but not sufficient.

The graph in the figure below is 2-connected but not Hamiltonian.

(ii) Page 287 lines 4 to 8

Consider the complete bipartite graph \( K_{m,n} \) with partite sets \( X \) and \( Y \). A Hamiltonian cycle in \( K_{m,n} \) must contain vertices alternatively from the partite sets \( X \) and \( Y \) since there are no edges joining
vertices in a partite set. Hence a necessary condition is \(|X| = |Y|\), i.e. \(m = n\). If this holds, starting from any vertex, it is easy to find spanning cycle in \(K_{m,n}\), since every vertex in \(X\) is adjacent to every vertex in \(Y\). So a complete bipartite graph is Hamiltonian only if it is of even order.

(iii)  **Page 287 lines 9 to 13**

The proposition states that for any Hamiltonian graph \(G\) and a non-empty set \(S \subset V(G)\), \(S\) has at most \(|S|\) components.

This fact will be more clear to you if you consider the following graph.

![Fig. 5](image)

Let us look at the proof.

Let \(G\) be a Hamiltonian graph of order \(n\) and let \(C = v_1, v_2, \ldots, v_n\) be a Hamiltonian cycle. Consider any non-empty subset \(S\) of \(V(G)\) and let \(G_1, G_2, \ldots, G_k\) be the components of \(G-S\).

Consider tracing along the Hamiltonian cycle \(C\) from vertex to vertex along edges joining them. To trace from one component to another we have to pass through \(S\). No pair of components has common vertices or edges joining the vertices in them. So, to trace from one component to another we have to pass through a vertex in \(S\) [various situations are shown in the figure]. But the constraints of Hamiltonian cycles prohibit passing through a vertex more than once. However, we have to trace all the components and back to the starting vertex. Thus, \(S\) must contain at least \(k\) vertices. This completes the proof.

But the condition given in the theorem is not sufficient. For example, consider the graph given in Fig. 1. It satisfies the condition of properties 7.2.3, but it is not Hamiltonian as there is no Hamiltonian cycle.

(iv)  **Page 287 lines 18 to 20**

Example 7.2.5 in *Textbook DW* gives two graphs. The first fails to satisfy the above condition because if deletion of the set (See Fig. 6 below) \(S = \{u, v\}\) from the graph results in three components. \(|S| < 3\), number of components.
The second graph (See Fig. 7 below) satisfies the condition, but not Hamiltonian.

This can be clarified in the following terms: There are three vertices of degree 2 in the graph and they are adjacent to the central vertex [the vertex of degree 6]. While tracing through the vertices of the graph for a Hamiltonian cycle, we need two visits at the central vertex more than once. This is not allowed in a Hamiltonian cycle. Hence, this graph is non-Hamiltonian.

(v)  Page 288 lines 1 to 3
The case of Petersen graph is also so.

(vi) Page 288 lines 4 to 13
This remark is about the strengthening of necessary condition to get sufficiency. Research on strengthening of the condition given in proposition 7.2.3 is going on and a brief history and a definition [toughness] are given in these lines.
Graph Theory

Now we make some general remark.

**Remark:** Whenever we have a necessary condition, we have to look whether it is sufficient or not. If not, strengthening of a necessary condition can be considered. In the case of Hamiltonian graphs, no nice necessary condition becomes sufficient but some sufficient conditions are arrived by strengthening necessary conditions or otherwise.

Why don’t you try some exercises now?

E1) Which of the following graphs are Hamiltonian? Give reasons for your answer. Also write down the Hamiltonian cycle wherever possible.

i) [Fig.9]

ii) [Fig.10]

iii) [Fig.11]
iv)

E2) Prove that the graph $K_{n,n}$ is Hamiltonian cycle and also write a Hamiltonian cycle in it.

The next section deals with sufficient condition.

## 10.3 SUFFICIENT CONDITIONS

Now we turn to sufficient conditions for a graph to be Hamiltonian. A characterisation of Hamiltonian graphs among a special class of graphs also is discussed in this section.

You can start reading the Textbook D.W.

READ TEXT BOOK DW Chapter 7, Section 7.2 pages 288 line 17 to page 290

NOTES:

(i) Page 288 lines 20 and 21

Here is a simple sufficient condition. If the minimum degree $\delta(G)$ is not less than half of the order of $G$ then the graph is Hamiltonian [Theorem 7.2.8 Textbook DW]. The bound is sharp in the sense that if the minimum degree is lower than half of order, then the graph need not be Hamiltonian. Families of such a graphs [counter examples] are described in Example 7.2.7 Textbook DW. The examples are obtained by ‘pasting’ two cliques at one of their vertices. The figure below shows such graph pasted at the vertex $v$. [a clique is a complete sub graph, see Definition 1.1.8 page 4 Textbook DW].

The following is a special case $n=11$
K₅ and K₆ pasted at v

This graph is not Hamiltonian since v is a cut vertex and
\[ \partial(G) = 4 = \frac{n(G)}{2} - 1 \]

This justifies the claim above since \( \partial(G) \) is just 1 less than half of the order of G, but G is non-Hamiltonian.

(ii) Page 288 lines 23 and 25

An example [corresponding to n = 1] of the graph described in Example 7.2.7 *Textbook DW* is given in the figure.

(iii) Page 288 lines 4 and 3 from below

This theorem due to Dirac proves the sufficiency of the condition. Here we shall give a full proof of the theorem.

**Proof:** Here we assume that \( n(G) \geq 3 \) for no graph of order 2 in Hamiltonian. The method of proof is assuming the other case and arrive at a contradiction. Suppose there is a non-Hamiltonian graph with the given conditions that there is a non-Hamiltonian graph G with \( \delta(G) \geq \frac{n(G)}{2} \). Let G be a maximal with their properties. That is if we add some edge to G, it will be come Hamiltonian. Hence, there is a Hamiltonian path \( v_1 - v_2 - \cdots - v_n \) with \( v_1 \) not adjacent to \( v_n \). Note that every spanning cycle is \( G + v_n v_1 \) containing the edge \( v_n v_1 \). Suppose there is an \( i \in \{2, 3, \ldots, n-1\} \) such that \( v_1 \) adjacent to \( v_{i+1} \) and \( v_n \) adjacent \( v_i \) is G.

Then \( v_1 - v_{i+1} - v_{i+2} - \cdots - v_n - v_{i-1} - v_{i+2} - v \), is a Hamiltonian cycle is G. So it is enough to prove such on \( i \) exist.

Let \( S = \{ i : v_1 \) is adjacent to \( v_{i+1} \} \)
Let \( T = \{ i : v_i \) is adjacent to \( v_i \} \)
Since \( \delta(G) \geq \frac{n(G)}{2} \), we have \(|S'|+|T'|\geq n\). Note that \( n \notin S \), for \( v_1 \) is not adjacent to \( v_1 \). Also \( n \notin T \). Hence \( n \notin S \cup T \).

But we have \(|S \cup T| = |S|+|T| - |S \cap T|\). Hence \(|S \cap T| \geq 1\) since \( S \) and \( T \) has a common element. Therefore there is an \( i \) such that \( v_i \) is adjacent to \( v_{i+1} \) and \( v_n \) is adjacent to \( v_1 \). Hence, there is a Hamiltonian cycle.

**(iv) Page 289 line 3**

Maximal non-Hamiltonian means that the graph becomes Hamiltonian if we introduce one edge joining any two non-adjacent vertices in it.

**(v) Page 289 line 6**

The symbol \( u \leftrightarrow v \) stands for ‘\( u \) is adjacent to \( v \)’. [See Textbook DW page 2 line 3 from bottom] and the symbol given here is the negation, \( u \) and \( v \) are not adjacent.

**(vi) Page 289 lines 8 to 9**

By the argument explained in note (iv), we have a Hamiltonian cycle in the graph obtained by adding a new edge in a maximal non-Hamiltonian graph. In the proof we construct a Hamiltonian cycle in the former graph using a Hamiltonian cycle in the latter.

**(vii) Page 289 lines 24 to 26**

The lemma 7.2.9 by Ore is a simple characterisation of Hamiltonian graphs among a special class of graphs.

Let us look at the details of the proof of Lemma 7.2.9.

**Proof lemma 7.2.9:** If \( G \) is Hamiltonian then it contains a Hamiltonian cycle and the cycle is a Hamiltonian cycle in \( G+uv \) also.

Conversely, let \( G + uv \) be Hamiltonian. Then \( G + uv \) contains a Hamiltonian cycle. If \( G + uv \) has a Hamiltonian cycle \( C \), not containing the edge \( uv \) then \( C \) is a Hamiltonian cycle in \( G \) also. Otherwise every Hamiltonian cycle in \( G + uv \) contains the edge \( uv \) and we can produce a Hamiltonian cycle from \( C \) as described in pages 7 to 18 page 289 Textbook DW.

This lemma is not applicable to all graphs since it demands a condition ‘\( d(u) + d(v) \geq n(G) \) for every non-adjacent vertices \( u \) and \( v \)’. This condition is not at all necessary for a graph to be Hamiltonian since every cycle is Hamiltonian but no cycle with more than four vertices possesses this condition.
Moreover, the conclusion depends on the Hamiltonicity of another graph. However, this gave rise to the concept of Hamiltonian closure, due to Bondy and Chvátal, of a graph defined in 7.2.10 page 289 *Textbook DW* and Theorem 7.2.11 on page 290 *Textbook DW*.

**(viii) Page 290 lines 3 and 4**

Theorem 7.2.11 requires no proof since it directly follows from lemma 7.2.9 and definition 7.2.10.

**(ix) Page 290 line 7**

This lemma assures the uniqueness of the closure of a graph.

Now we make a remark.

**Remark 1**: If the condition that \( d(u) + d(v) \geq n(G) \) is dropped, then \( G \) need not be Hamiltonian even if \( G + uv \) is Hamiltonian.

The following graph illustrates this.

![Fig. 15](image)

Here \( G + uv \) is Hamiltonian for any two non-adjacent vertices in \( G \), but this not Hamiltonian.

**Example 5**: Let us find the closure of the graph given below:

![Fig. 16](image)
Let us denote the given graph by $G_1$. Then the closure is given by the following iteration:

i) Ist iteration

![Fig17]

ii) IInd Iteration

![Fig18]

iii) IIIrd Iteration

![Fig19]

Hence $G_4 = C(G) = K_8$
You can try some exercises now.

E3) Show that if the closure of a graph is complete, then it is Hamiltonian. Show that the converse is not true.

E4) Use Ore’s theorem to show that the following graph is not Hamiltonian.

![Fig.20](image)

10.4 SUMMARY

In this unit, we have studied

1) concepts of Hamiltonian cycles and Hamiltonian graphs.

2) two necessary conditions for graph to be Hamiltonian.

3) two sufficient conditions for a graph to be Hamiltonian.

4) two characterizations of Hamiltonian graphs.

5) concept of the closure of a graph.

10.5 HINTS/SOLUTIONS

E1) i) This is a Hamiltonian cycle. The cycle is given by $v_1 - v' - v_2 - v_3 - v' - v_4 - v_5 - v_1$.

ii) Not Hamiltonian-since it is a bipartite graph with odd number of vertices.

iii) The Not Hamiltonian since this is a true and therefore has no cycle.

iv) This is Hamiltonian-the cycle is given by $v_1 - v_2 - v_3 - v_4 - v_5 - v_1$.

E2) Let $x = \{u_1, u_2, \ldots, u_n\}$, $y = \{v_1, v_2, \ldots, V_n\}$ be the bipartition of $K_{n,n}$. Then the cycle $u_1-v_1-u_2-v_2, \ldots, u_n-v_n-v_1$ is a Hamiltonian cycle.

E3) If the closure of a graph G is complete, then the closure is Hamiltonian because any complete graph is Hamiltonian. Therefore G is Hamiltonian. The converse of the statement is that if the closure of a
graph is Hamiltonian, then the graph is complete. This is not-true for a counter example, any the cyclic graph with more than four vertices is Hamiltonian and is its own closure and therefore is not complete.

E4) We note that \( d(u) + d(v) \geq 5 \) for each pair of non-adjacent vertices \( u \) and \( v \). Therefore by Ore’s theorem the given graph is Hamiltonian.
APPENDIX – PRACTICAL SESSIONS

As we have mentioned in the course introduction, you need to gain some practical experience of some of the algorithms learnt in this course. Keeping in view of this, we have included a practical component for this course. You will recall that in the 70% marks earmarked for the term end examination, 50% is for the theory and 20% is for the practical. In the 30% marks for continuous assessment, 20% is of the assignment and 10% is for the practical component.

This course has 8 practical sessions of 3 hours each. The language to be used for writing the programs is C. You are already familiar with C-programming from the course MMT-001. Apart from this, the course MMTE-002, which you will studying along with this course also will help you to write the programs.

While writing the program, you should add comments in the program and in the functions, documenting what the program or function does. Indent your programs properly so that anyone can read them easily and understand them. If you are working on Linux, the indent program may be available on your computer. For example, if you want to indent your program in the style followed in the book by Kernighan and Ritchie, you can invoke indent using the command:

```
indent –kr file_name.c
```

Indent offers many other options. Read the documentation for the program if necessary.

Also, write a friendly interface, wherever required, for the user to interact with the program. Your program should also check if the input is appropriate and print an error if it is not.

Now we list the practical sessions for this course. For each of the problems given below for practical work you are expected to do the following:

1. Write the algorithm in pseudo code or make a flow chart and show it to the counselor.
2. After the counselor approves, implement the algorithm in C-language, i.e. write a C-program, debug it, compile it and sum it for the sample graph given and show it to the counselor.
3. The counselor may ask you questions about the program to test your understanding and also may suggest changes.
4. After the counselor approves the program, print or write neatly the source code of the program and get it signed by the counselor.
5. Maintain your program in a file. Your continuous assessment will be based on this file.
6. You should also produce the file at the time of practical examination.
Now we list the practical sessions for this course.

**Session 1:**

The aim of this session is to give you some experience of how to write programs for some of the basic concepts of Graph Theory. As a first step we consider how to represent a graph as an object in computer programming.

The most common way to represent a graph is by its adjacency matrix. With an adjacency matrix, we can determine immediately whether or not there is an edge from vertex i to vertex j, just by checking whether column j and row i and of the matrix is nonzero. For the undirected graphs that we are considering, if there is an entry in row i and column j, then there also must be an entry in row j and column i, so the adjacency matrix is symmetric in this case. Fig.1 shows an example of an adjacency matrix for an undirected graph; we have discussed adjacency matrix in Unit 1, Block 1 of this course.

```
0 1 1 0 0 1 1 1
0 1 1 0 0 0 0 1
1 0 1 0 0 0 0 1
2 0 0 1 1 1 1 0
3 0 0 1 1 0 0 0
4 0 0 0 1 1 1 1 0
5 1 1 1 0 1 0 0 1
6 1 1 1 0 1 0 0 1
7 1 1 1 0 1 0 0 1
```

Fig.1

Another straightforward method for representing a graph is to use an array of linked lists, called adjacency lists. We keep a linked list for each vertex, with a node for each vertex connected to that vertex. For the undirected graphs that we are considering, if there is a node for j in i’s list, then there must be node for i in j’s list. Fig. 2 shows an example of the adjacency-list representation of an undirected graph given in Fig.1.
Now you can do the following:

**Program 1:**

i) Write a program that prints the adjacency matrix for a graph, given a sequence of edges as input.

ii) Use the program to find the adjacency matrix for the graph given in Fig.3.

![Fig.3](image)

**Program 2:**

i) Write a program that prints the adjacency list for a graph. [You may like to refer Unit 7, Block 2, MMT-001 and revised the material on bounded list.]

ii) Use the program to find the adjacency lists for the graph in Fig.3

**Session 2:**

Here we consider incidence matrix which is another representation for a graph.

Let G be an undirected graph. If we denote the vertices by $v_1, \ldots, v_p$ and the edges by $e_1, \ldots, e_q$, then incidence matrix $B$ of $G$ is the $p \times q$ matrix whose entries are given by

$$B(i, j) = \begin{cases} 1 & \text{if vertex } v_i \text{ is incident with edge } e_j, \\ 0 & \text{otherwise} \end{cases}$$

**Program 3:**

i) Write a program that constructs and prints the incidence matrix of a given undirected graph. [Refer Unit 7, Block 2, MMT-001 and revise the material on arrays.]

ii) Use the program to find the incidence matrix of the graph given in Fig.1.
**Program 4:**

Use the program to verify the relationship $BB^T = A + D$ for any Graph $G$ where $B$ is the incidence matrix of $G$, $B^T$ denotes its transpose, $A$ is the adjacency matrix and $D$ is the diagonal matrix which is defined as $|V| \times |V|$ matrix whose diagonal entries are the degrees of the vertices of $G$ and whose off diagonal entries are all zero.

**Session 3: Kruskal’s Algorithm**

In this section we want you to implement Kruskal’s algorithm that we have discussed in Unit 5 Block 1 of the Study Guide.

**Program 5:**

i) Write a program that uses Kruskal’s Algorithm (see Textbook D.W. page 95), to find a minimum spanning tree for a weighted connected graph.

ii) Using the program find a minimum spanning tree for the connected graph given in Fig.3.

**Session 4: Breadth First Search Algorithm**

Breadth First Search algorithm finds the length of a shortest path from the vertex start to every other vertex in a graph with vertices 1,2,...,n. The graph is represented using adjacency lists; adj[i] is a reference to the first node in a linked list of nodes representing the vertices adjacent to vertex i. Each node has members ver, the vertex adjacent to i, next, a reference to the next node in the linked list or null, for the last node in the linked list.

In the array length, length[i] is set to the length of a shortest path from start vertex to vertex i if this length has been computed or $\infty$ if the length has not been computed. If there is no path from start vertex to vertex i, when the algorithm terminates, length[i] is $\infty$.

The diameter of a tree $T$ with vertex set $V(G)$ is given by $\max_{u,v} \delta(u,v)$, where $\delta(u,v)$ denotes the shortest-path distance between $u$ and $v$. That is, the diameter is the largest of all the shortest-path distances in the tree.

**Program 7:**

i) Write a program that uses Breadth-First Search-(BFS) algorithm (page 99 Textbook DW), to find the diameter of a given tree $T$.

ii) Using the program find the diameter of the tree given on the next page (See Fig.4).
Sessions 5: Dijkstra’s Algorithm

Here we consider Dijkstra’s algorithm discussed in Unit 5, Block 1. This algorithm finds shortest paths from the designated vertex “start” to all of the other vertices in a connected, weighted, n-vertex graph. The graph is represented using adjacency lists; adj [i] is a reference to the first node in a linked list of nodes representing the vertices adjacent to vertex i. Each node has members ver, the vertex adjacent to i; weight representing the weight of edge (i, ver); and next, a reference to the next node in the linked list or null, for the last node in the linked list. In a shortest path, the predecessor of vertex i ≠ start is predecessor [i], and predecessor [start] = 0.

Program 6:

i) Using Dijkstra’s Algorithm (Textbook DW page 97), write a program to find the shortest path between a pair of vertices x and y in a weighted graph G.

ii) Using the program find the shortest path between A and D in the graph shown in Fig.3.

Session 6: Bipartite Graph Algorithm

You are already familiar with Bipartite Graph (See Unit 2 of Block 1 of Study Guide). You are also familiar with the Breadth First Search algorithm for finding the shortest path. Here we want you to extend this algorithm to check whether the graph is bipartite or not. We assume that the graph is connected.

We modify the BFS algorithm as follows:

When you reach a vertex, check whether you have already reached before. If you have already reached before as in BFS, we just record the length of the shortest path. If you have already reached the vertex before, check whether the length of the current path and the shortest distance already recorded earlier have the same parity (odd/even). If they are of the same parity proceed further as you usually do in BFS. If they are not, then the graph has odd cycle and therefore is not bipartite. So you can terminate the algorithm with the output that the graph is not bipartite. If you have successively traversed all the vertices, then the graph is bipartite.
**Program 8:**

i) Write a program to check whether a graph is Bipartite or not.

ii) Using the program check whether the following graph is bipartite or not.

![Graph](image)

**Fig. 5**

**Session 7: Depth First Search (DFS) Algorithm**

Depth First Search (DFS) algorithm is useful

- to find a path from one vertex to another
- to check whether or not graph is connected
- for finding a spanning tree of a connected graph

This algorithm gives a procedure for systematically visiting all the vertices in a graph. It is a simple recursive algorithm which involves the following procedure.

i) start with any vertex \( v \)

ii) visit \( v \)

iii) (Recursively) visit each (unvisited) vertex attached to \( v \).

If the graph is connected we eventually reach all the vertices.

We shall illustrate the algorithm for the graph given in Fig.1 (see Fig.1) whose adjacency list is given in Fig.2.

The sequence given below shows the sequences of function visits that constitutes Depth-first search.

**ALGORITHM: Depth-first-search**

visit 0

visit 7 (first on 0’s list)

visit 1 (first on 7’s list)
check 7 on 1’s list
check 0 on 1’s list
visit 2 (second on 7’s list)
    check 7 on 2’s list
    check 0 on 2’s list
    check 0 on 7’s list
visit 4 (fourth on 7’s list)
    visit 6 (first on 4’s list)
        check 4 on 6’s list
        check 0 on 6’s list
visit 5 (second on 4’s list)
    check 0 on 5’s list
    check 4 on 5’s list
visit 3 (third on 5’s list)
    check 5 on 3’s list
    check 4 on 3’s list
    check 7 on 4’s list
    Check 3 on 4’s list
check 5 on 0’s list
check 2 on 0’s list
check 2 on 0’s list
check 6 on 0’s list
The figure (Fig.7) given on the next page depicts the way in which the edges in the graph are followed.
Note that each edge in the graph is followed with one of the two possible outcomes. If the edge takes us to a vertex that we have not yet visited, we follow it there via a recursive visit. The set of all edges that we choose in this way forms a spanning tree as shown in Fig. 7.
Program 9:

i) Write a program that uses Depth first search algorithm to find path between vertices x and y of a Graph G. Also write another program to check whether a graph is connected or not.

ii) Using the program find the path between vertices A and D in Fig.3. Also check the connectivity of the graph in Fig.3.

Session 8: Ford-Fulkerson Labeling Algorithm

You are already familiar with Ford-Fulkerson labeling algorithm for finding maximum flow from Unit 7 [see page 179 Text Book DW], Block-2 of Study Guide.

Program 10:

i) Write a program that Uses Ford-Fulkerson labeling algorithm (Textbook D.W. – page 179), to find the maximum flow between two vertices s and t in a network.

ii) Using the program find the maximum flow from s to t in the graph given in Fig. 8.

![Fig.8](image_url)