Department of Mathematics.

SEM - 4

Course - BMH4SEC21.

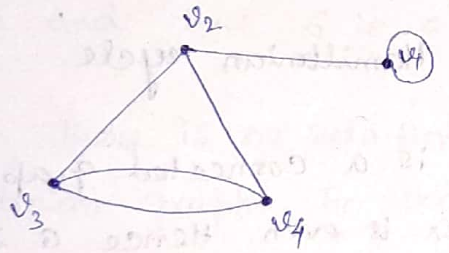
Graph Theory

Notes given by Rima Dutta.

Matrix Representation of a graph.

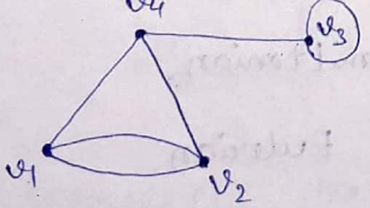
Definition (Adjacency matrix) :- Let G be a graph with n vertices, listed as $v_1, v_2, \ldots, v_n$. The adjacency matrix of G, with respect to this particular listing of vertices of G, is the $n \times n$ matrix $A(G) = (a_{ij})_{n \times n}$, where the (i,j) th entry a_{ij} is the number of edges joining the vertex v_i to the vertex v_j.

Ex. Consider the following graph.



Then the adjacency matrix is $A(G) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 2 & 1 \end{bmatrix}$

2. Note :- 1. The matrix can be changed if the labelling the vertices is changed. For example,



then $A(G) = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

2. Adjacency matrix must be square matrix.

3. For any adjacency matrix, $A(G) = (a_{ij})_{n \times n}$ of a graph G, we have a_{ij} = the number of edges joining the vertex v_i to the vertex v_j = the number

of edges joining the vertex v_j to the vertex $v_i = a_{ji}$.

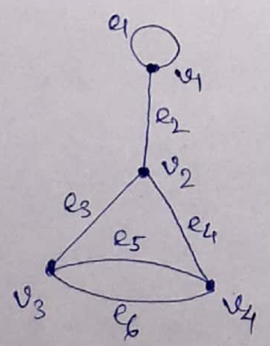
Thus the adjacency matrix is symmetric matrix.

4. If all diagonal elements of a adjacency matrix are zero, then the graph does not contain any loop. Also if off diagonal contains other than 0 and 1, then the graph must contain parallel edges.

<u>Definition</u> (<u>Incidence matrix</u>):- Let G be a graph with n vertices listed as $v_1, v_2, \dots, v_n$ and q edges, listed as $e_1, e_2, \dots, e_q$. Then the ~~end~~ incidence matrix of G, with respect to those particular listing of vertices and edges of G, is the $n \times q$ matrix $M(G) = (m_{ij})_{n\times q}$ where m_{ij} is the number of times that the vertex v_i is incident with the edge e_j ie.

$$m_{ij} = \begin{cases} 0 & \text{if } v_i \text{ is not an end of } e_j \\ 1 & \text{if } v_i \text{ is } \cancel{\text{not}} \text{ an end of the non-loop } e_j \\ 2 & \text{if } v_i \text{ is an end of the loop } e_j. \end{cases}$$

<u>Ex.</u> Consider the following graph and its incidence matrix.



$$M(G) = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Note :- 1. Incidence matrix is a rectangular matrix.

2. The sum of the entries in the ith row of M(G) the incidence matrix of a graph G, gives us the degree of the vertex v_i while the sum of the entries in each column of $M(G) = 2$.

3. Incidence matrix may not be a symmetric matrix.

4. Entries of adjacency matrix are 0, 1, 2, 3, ... but the entries of an incidence matrix are either 0 or 1 or 2.

5. ~~Each row sum of the incidence matrix is the degree of the vertex v_i, column sum is~~

Travelling salesman problem :

The travelling salesman problem is stated as follows :

A salesman is required to visit a number of cities during a trip. Given the distances between the cities, in what order should he travel so as to visit every city precisely once and return home, with the minimum mileage travelled?

This problem ~~is~~ can be expressed by a graph as follows:

Represting the cities by vertices and the roads between them by edges, we get a graph. In this graph, with every edge e_i, there are associated a real number $w(e_i)$. Such a graph is called a weighted graph, $w(e_i)$ being the weight of edge e_i.

The total number of different Hamiltonian circuits in a complete graph with n vertices can be shown to be $(n-1)!/2$. [This follows from the fact that starting from any vertex we have $(n-1)$ edges to choose from the first vertex, $(n-2)$ from the second, $(n-3)$ from the third and so on. These being independent choices, we get $(n-1)!$ possible number of choices. This number is divided by 2, because each Hamiltonian circuit has been counted twice].

This problem can always be solved by enumerating $(n-1)!/2$ Hamiltonian circuits, calculating the distance travelled in each, and then picking the shortest one.

The problem is to require a managable algorithm for finding the shortest route. But no efficient algorithm for problems of arbitrary size has yet been found.

~~Dij~~ Dijkstra's Algorithm :-

Dijkstra's algorithm is use to find the shortest path between two vertices in a graph.

There are some steps for finding shortest path by using this algorithm.

Step 1:- Remove all the loops

Step 2:- Remove all the parallel edges between two vertices except the one with least weight. (Weight are given in the graph).

Step 3:- Create the weight matrix table

(i) Set 0 to the source vertex and infinite to the remaining matrix.

~~(ii)~~ For all vertices, repeat (ii) and (iii)

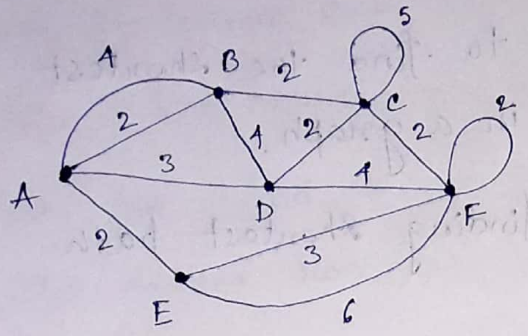
(ii) Mark the smallest unmarked value and mark that vertex.

(iii) Find those vertices which are directly connected with marked vertex and update all.

Update value formula :-

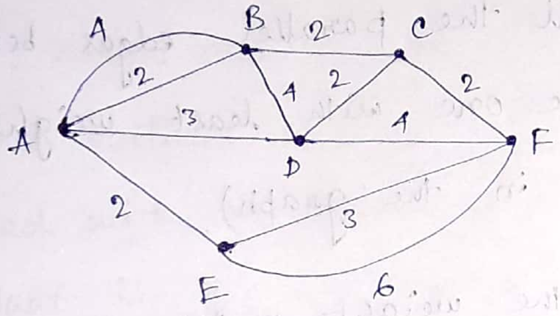
New destination value = minimum (Old destination value, Marked value + Edge Weight).

Ex.

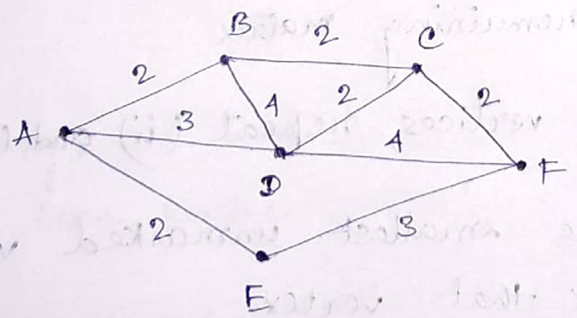


The digits written in the graph are Weight. ~~destination value~~ / ~~weight~~.

Step 1 :-



Step 2 :-



step 3 :- Suppose source vertex is A. and destination vertex F. Now we find out the shortest path from A to F.

For this we develop the weight matrix.

Marked vertex	A	B	C	D	E	F
A	$\boxed{0}$	∞	∞	∞	∞	∞
B	0	$\boxed{2}$	∞	3	2	∞
E	0	2	4	3	$\boxed{2}$	∞
D	0	2	4	$\boxed{3}$	2	5
C	0	2	$\boxed{4}$	3	2	5
F	0	2	4	3	2	$\boxed{5}$

First, the value of A in the first row is 0, that is the smallest value in that row and hence we ~~2nd~~, mark this value.

2nd, the vertices B, D, E are directly associated with the vertex A.

So, the value of B, D, E should be updated.

∴ Updated value of B

= min (Old destination value, marked value + Edge weight)

= min (∞, 0+2)

= min (∞, 2) = 2.

So the destination value of B = 2 and we write 2 in the 2nd row and B column.

Similarly, updated value of D and E are 3 and 2 respectively.

Scanned with CamScanner

Again, similarly, we write 3 and 2 in the 2nd row and D column and E column.

Also, other remaining column remains unchanged.

Now, ~~the~~ find the smallest destination value in the second row, and that is 2. So, we marked the value 2.

Continue the ~~one~~ same process for all the remaining rows, and ~~in~~ here we update the value of the vertices which are directly ~~att~~. attached with marked vertex in the previous row and that is B.

This process will be continued for all the rows.

In this way, we develop the weight matrix, which has been written in the previous page.

Therefore the shortest path is given by

$F \leftarrow E \leftarrow A$.

[The elements in the weighted matrix is called destination matrix].