Aninesh Thomas
22PMC113

Graph Theory Problems

1) To prove the sum of degree of all vertices in a graph G is even, The sum of all vertices of a graph is twice the number of edges in the graph

ie $\sum_{i=n} \cancel{deg(v_i)} =$

$$\Sigma deg(v) = 2|E|$$

where deg(v) is the degree of vertex v, |E| is the no of edges in the graph

Now if we consider sum of the degrees of all vertices in the graph G, we have

$\Sigma deg(v) = 2|G|$

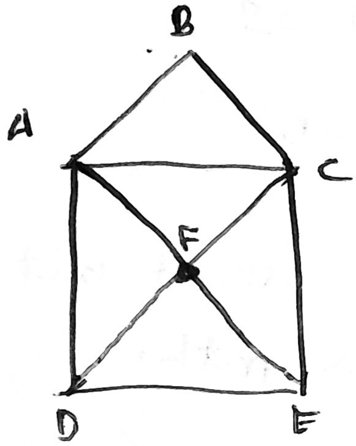
since $2[E]$ is even, it follows that the sum of the degree of all vertices in the graph is also even, Therefore, we have proved that the sum of the degree of all vertices in graph G is even.

If each 9 of the mathematicians shake hands every with exactly 7 people at the seminar, then the total no of handshake would be

$\frac{9 \times 7}{2} = 31.5$ which is not a complete number.

However, since a shakehand involves two people, the total no of handshakes must be an integer, therefore, it is not posible for each mathematician to have shaken hands with exactly 7 people at the seminar.

2) 1) Graph - I



deg(A) = 4   deg(E) = 3

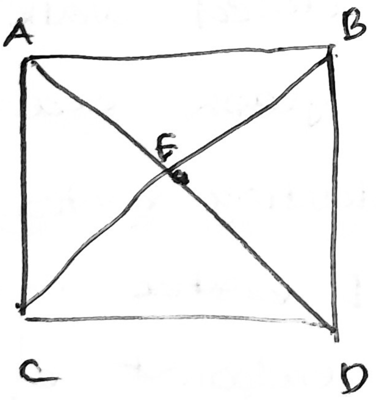
deg(B) = 2   deg(F) = 4

deg(C) = 4

deg(D) = 3

This is not a euler circuit, But this graph is having euler path.

Graph 2

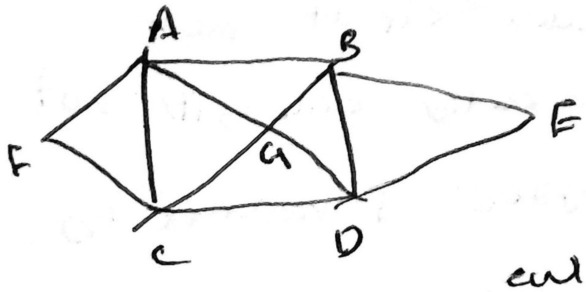


degree of A = B = C = D = 3

degree of E = 4

Not having euler path or circuit In this graph

Graph 3



deg (A, B, C, D) = 4

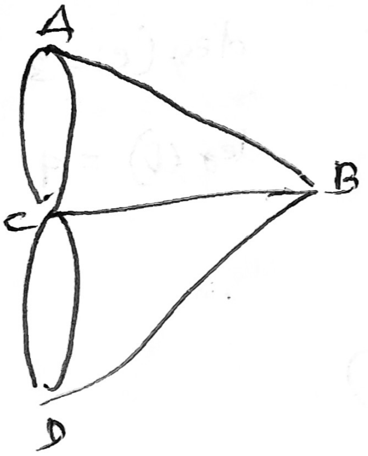
deg(E) = C(F) = 2

degree of G = 4

euler circuit, but not path

~~Not having~~ circuit or path

Graph 4



deg (A, B, C) = 3

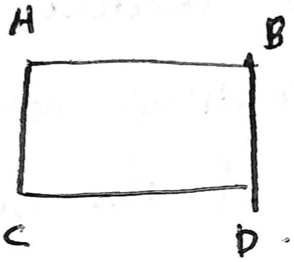
deg of D = 5

No euler circuit but having euler path.

2,2. No it is not posible for a graph with a degree 1 vertex to have an euler circuit. An euler circuit is a closed walk that visit every edge of a graph exactly once and return to starting vertex. If a graph has degree 1 vertex that vertex would be an endpoint of an edge and would need to be visited twice in order to form a closed walk. However since an euler circuit must visit every edge exactly once, its not posible for a degree 1 vertex to be part of an euler circuit.

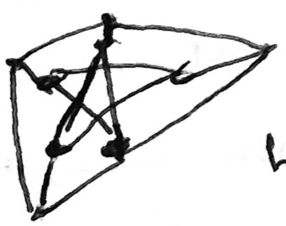
2 c3) If every vertex of a graph has deg ole, then the graph has both euler path & circuit, All vertices of even degree no one has odd degree

eg.

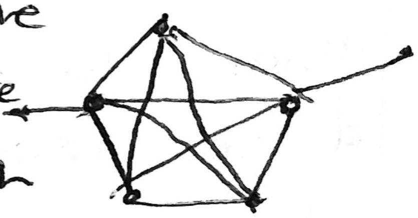


2 c4) An euler path exists in a graph if and only if the graph has exactly zero or two vertices with odd degree. An euler circuit exists in a graph if and only if all vertices have even degree

3) The graph on the left side has an hamiltonian path.



the graph on the side doesn't have hamiltonian path



5 Dijkstra's Algorithm

It allows us to find the shortest path b/w any two vertices of a graph. It differs from the minimum spanning tree because the shortest distance b/w two vertices might not include all the vertices of the graph,

Steps

1). Make all the nodes unvisited
2) Mark the picked starting node with a current distance of 0 and the rest nodes with infinity
3) Now fix the starting node as the current node
4) For the current node, Analyze all of its unvisited neighbours & measure their distance by adding the current distance of the current node to the weight of the edge that connects the neighbour node & current node.

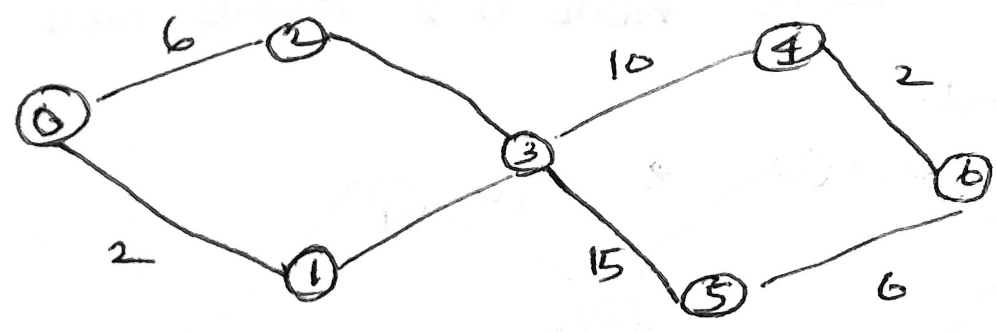
5) compare the recently measured distance with the current distance assigned to the neighbouring node & make it as the new current distance of the neighbouring node.

6) After that, consider all the unvisited neighbours of the current node, mark the current node as visited

7) If the destination node has been marked visited then stop, an has end - end

8) else, choose the unvisited node ie, marked with the least distance, fix it as the new element node and repeat the process again from step 4.

Example



for this graph, we will assume that the weight of the edges represent the distance b/w two nodes.

we have shortest paths from node 0 to 1, 0→2, 0→3, 0→4, 0→5 and 0→6.

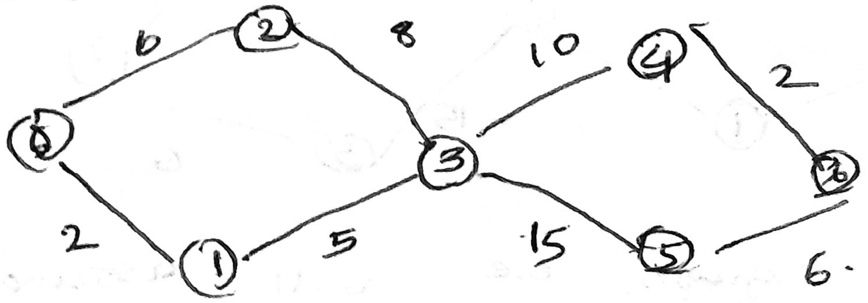
The distance from the source node to itself is 0, In this, source node is 0

The distance from source to all other nodes is unknown, so mark them as unknown ie, 0→0, 1→∞, 2→∞, 3→∞, 4→∞, 5→∞, 6→∞

Algorithm will complete when all the nodes marked as visited to the distance b/w them added to the path, unvisited nodes : 0 1 2 3 4 5 6

<u>Step 1</u>

start from node 0 & mark node as visited.



unvisited nodes. = {1, 2, 3, 4, 5, 6}

Distance: 0 : 0

1 : α

2 : α

3 : α

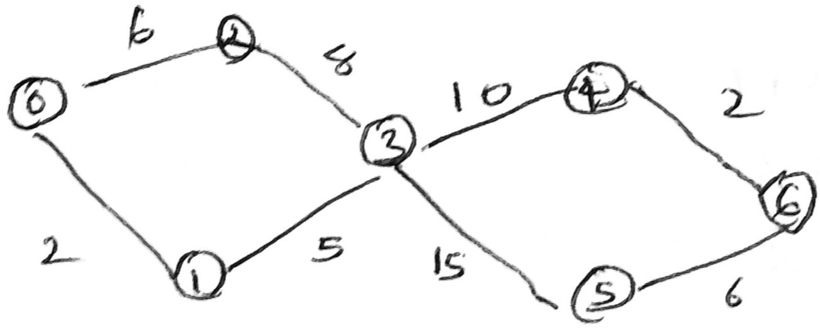
4 : α

5 : α

6 : α

step 2

mark node 1 as visited and add the distance



unvisited node {2, 3, 4, 5, 6}

Distance:

0 : 0   4 : α

1 : 2   5 : α

2 : α   6 : α

3 : α

step3

mark node as visited after considering the optimal paths and add the distance

unvisited nodes {4, 5, 6}

0: 0
1: 2
2: 6
3: 7
4: 2
5: α
6: α.

step4

unvisited nodes = {5, 6}

0: 0
1: 2
2: 6
3: 7
4: 11
5: α
6: α.

steps

~~unvisited nodes {6}~~

move forward & check the adjacent node ie, node 6 so mark it as visited and add up the distance.

unvisited node {5}

0 : 0   4 : 17
1 : 2   5 : 22
2 : 6   6 : 19.
3 : 7

0 -> 1 -> 3 -> 4 -> 6 ie, = 2+5+10+2

so shortest distance from = 19.
source vertex is 19 optimal.

ii) The Number of different trees that can be drawn 4 vertices is 2.
The Number of different trees can be draw 5 vertices is 3.

7) ~~Greedy~~ Kruskal's Algorithm is a greed algorithm used to find min spanning Tree (MST) of a weighted undirected graph, the MST of a graph is a tree that spans all the vertices of the graph, with minimum total edge weight

Steps

1) Sort the edges of the graph is non

decreasing order of their weights.

2) Initialize a set for MST

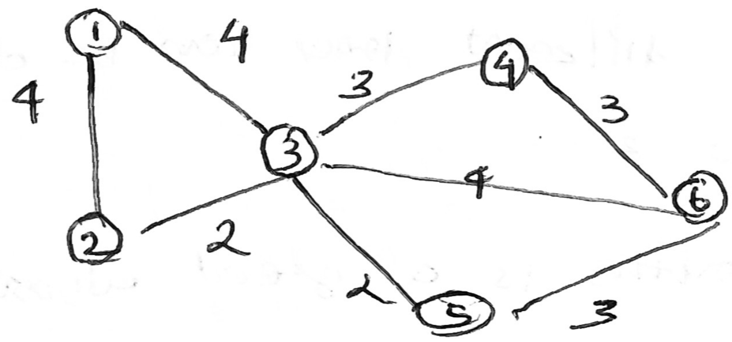
3) For each edge in the sorted list of edges

If the edge doesn't create a cycle in the MST set add it to set

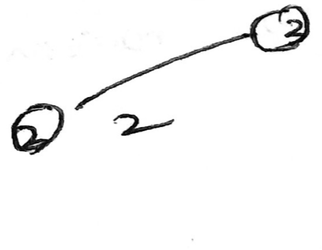
If the edge creates a cycle, discard it.

4) Stop when all vertices have been included in the MST set

Example



start with the weighted graph

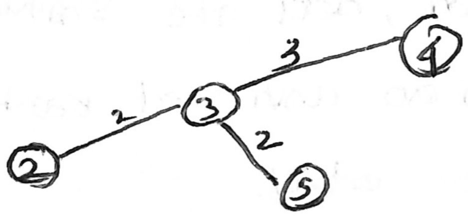


Step2

choose the edge with the least weight, if there are more than 1, choose anyone.

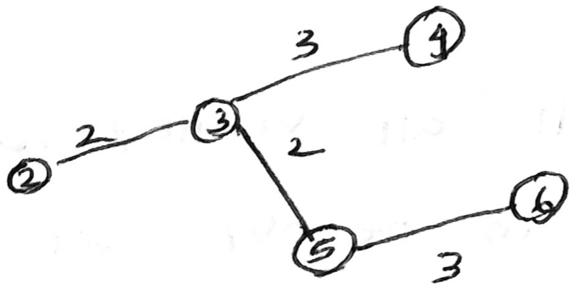
Step3

choose the next shortest edge and add it.

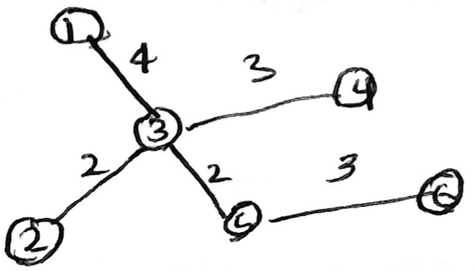


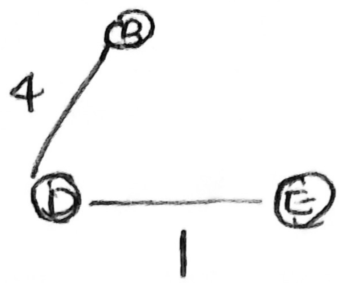
Step4

choose the next shortest edge and add it.

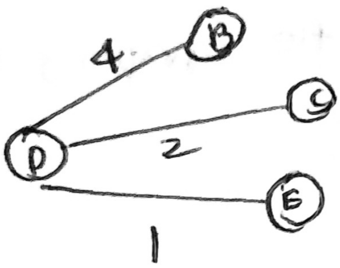


Step5



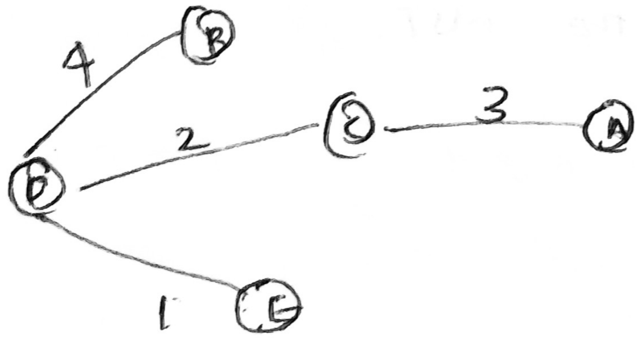


Step4
Select the edge CD and add it to MST.



Step5

Now choose the edge CA, here we cannot select the edge CE as it would create a cycle, to the graph so choose the edge CA & add it to MST



so the graph produced in the step 5 is the spanning tree of the given graph ∴ cost = 4+2+3+1 = 10 units.

9) Binary Tree

An ordered rooted tree whose subtree are put into a delenite order & are themselves ordered rooted trees.

A Tree consenting of no vertices is a binary Tree.

A vertex together with two subsets are both binary trees is a binary tree. The subtree is called the left & right subtree of the binary Tree.

10) In a complete binary Tree there exactly 2^0 =1 root, ie, at level 1, the next level contains 2 vertices. The next level contains 2^2 =4 vertices. thus the total no of vertices in a complete binary Tree of level n is 1+2+4+8+ .... +2^(n-1) = 2^n-1

∴ the total is always an odd number

Scanned with ACE Scanner

11) prove that $p = \frac{n+1}{2}$

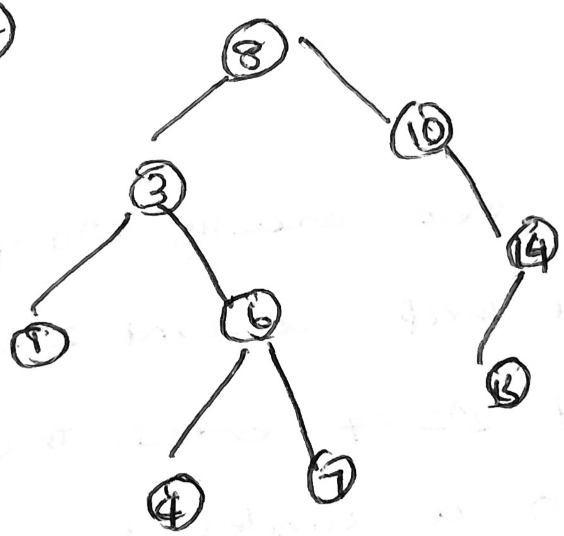
let p be the no of pendent vertices in a binary Tree T then "n-p-1" is the no of vertices of degree.

∴ no of edges in T equals.

$\frac{1}{2}[p+3(n-p-1)+2] = n-1$

hence $p = \frac{n-1}{2}$

12)



preorder = 8 3 1 6 4 7 10 14 13

Inorder = 13 4 6 7 8 10 13 14

postorder = 14 7 6 3 13 14 10 8

Scanned with ACE Scanner