DSA

23221

Page No. 1
Date.

Assignment No. 07

Minimum Spanning Tree

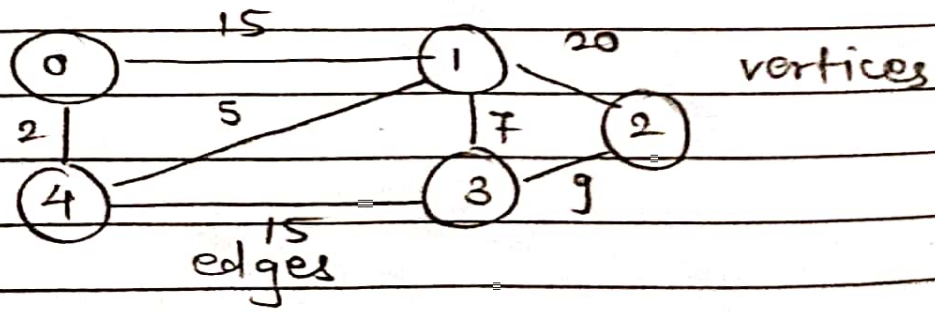
Aim : To implement minimum spanning tree using prims and kruskals algorithm

Problem Statement: Represent graph of your college campus using adjacency list/matrix. Nodes represent various departments and link should represent the distance betn them find minimum spanning tree using

a) Kruskal's Algorithm
b) prim's algorithm
Analyse time & space complexity

Theory:

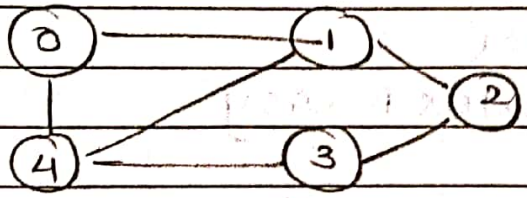
1) **Graph:**
It is an non-linear data structure consisting of nodes and edges



Page No.
Date.

e) Graph representation using 2D Array Matrix

- Adjacency matrix is a 2D Array of size V x V where V is the number of vertices in graph.
- let 2D Array be adj[][], then adj[i][j] = 1 if there is an edge from V to V2 i to j.
- For undirected graph matrix is symmetric



Adjacency matrix:

	0	1	2	3	4
0	0	1	0	0	1
1	1	0	1	1	1
2	0	1	0	1	0
3	0	1	1	0	1
4	1	1	0	1	0

Advantages:

1) Easier representation
2) Removing an edge takes O(1) th time

Disadvantages:

1) Consume more space
2) More time to add an vertex

Graph representation using adjacency list:

- An array of list is used.
- Size is equal to no of vertex.
- Let arr[] be an array. An entry arr[i] represents list of vertices adjacent to vertex

Advantages:

1) Saves space
2) Adding vertex is easy

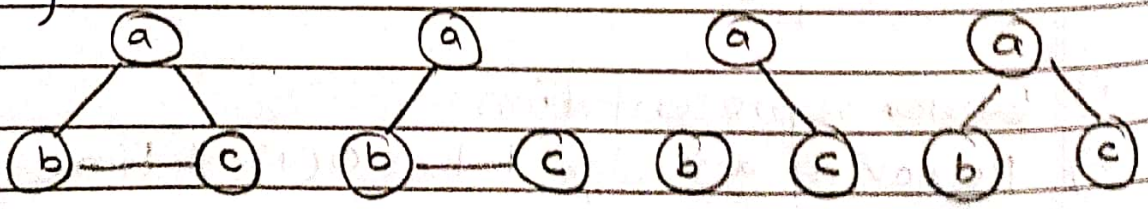
Application of graph:

1) Maps
2) Computer Network
3) Social networking sites
4) To represent flow of computations

Spanning Tree:

A spanning tree is subset of Graph G, which has all the vertices covered with minimum possible no of edges

eg



Minimum Spanning Tree:

A minimum spanning tree is spanning tree with weight less than or equal to weight of every other spanning tree. The weight is sum of weight on each edge of spanning tree.

Page No.
Date.

Uses of MST:

1) Telephone wiring
2) Electronic circuits
3) Computer Networks

Algorithms:

1) prim's algorithm
- It starts with tree, T, consisting of single vertex v.
- Then it find shortest edge emanating from x that connects T to rest of the graph
- It adds this edge & new vertex to tree T
- It then picks the shortest edge emanating from the revised tree T that also connects T to the rest of graph & repeats the process till all vertex are visited

2) Kruskal's algorithm
- Sort all edges is ascending order of weights.
- pick the smallest edge check if it forms cycle with spanning tree. If not include this edge else discard it.
- Repeat (b) till there are (v-1) edges

validations:

- No of vertices and edges should be positive
- start & end vertices are within the no of vertices provided by user

Test cases :

- Complete undirected graph with no loops, parallel edges.
- Connected undirected graph with no loops, parallel edges.

Conclusion :

- Time complexity of prim's algorithm $O(V^2)$ for adjacency matrix
  For adjacency list = ~~o(e~~ $O(E \log V)$
  It works faster with dense graphs
- Time complexity of kruskal's algorithm $O(E \log e)$ or $O(E \log V)$

  It runs faster in sparse matrix.