DATA STRUCTURES AND ALGORITHM

Module

8

- To understand application of a graph theory
- To apply the concept of Depth First Search and Breadth First Search

DEFINITION

A **Minimum Spanning Tree (MST)** is a subgraph of an undirected graph such that the subgraph spans (includes) all nodes, is connected, is acyclic, and has minimum total edge weight

Algorithm Characteristics

Both Prim's and Kruskal's Algorithms work with undirected graphs

Both work with weighted and unweighted graphs but are more interesting when edges are weighted

Both are greedy algorithms that produce optimal solutions

Kruskal's Algorithm (A-Z)

Work with edges, rather than nodes

Two steps:

- Sort edges by increasing edge weight
- Select the first $|V| - 1$ edges that do not generate a cycle

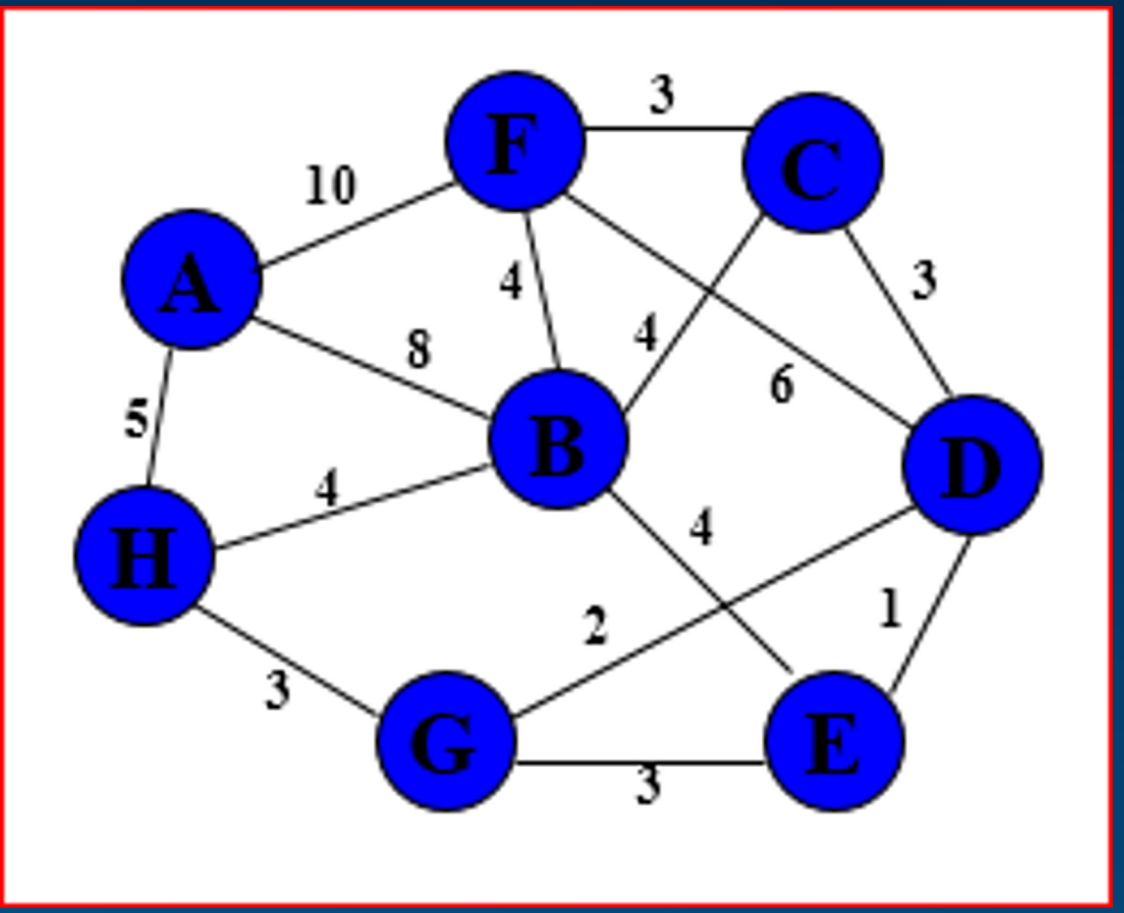
Kruskal's Algorithm

ALGORITHM

Suppose G = (V, E) is a graph, and T is a minimum spanning tree of graph G.

1. Initialize the spanning tree T to contain all the vertices in the graph G but no edges.
2. Choose the edge e with lowest weight from graph G.
3. Check if both vertices from e are within the same set in the tree T, for all such sets of T. If it is not present, add the edge e to the tree T, and replace the two sets that this edge connects.
4. Delete the edge e from the graph G and repeat the step 2 and 3 until there is no more edge to add or until the spanning tree T contains (n-1) vertices.
5. Exit

Kruskal's Algorithm



Sort the edges by increasing edge weight

Chronological order

x x x x

(D,E)	1	T
MST		

Prims's Algorithm

This algorithm was discovered by Vojtech Jarnik in 1936 and later rediscovered by Robert Prim. Prim's algorithm also constructs the minimum-cost spanning tree, edge by edge. Prim's algorithm begin with a tree T that contain a single vertex (this vertex can be of any vertices in the original graph), generally it is selected as lower most cost edge in the tree. Then we add a least cost edge (u, v) to T such tat T U {(u, v) } is also a tree. Repeat this edge-addition step until T contains n-1 edges.

Prims's Algorithm

ALGORITHM

Suppose G = (V,E) is a graph and T is a minimum spanning tree of graph G.

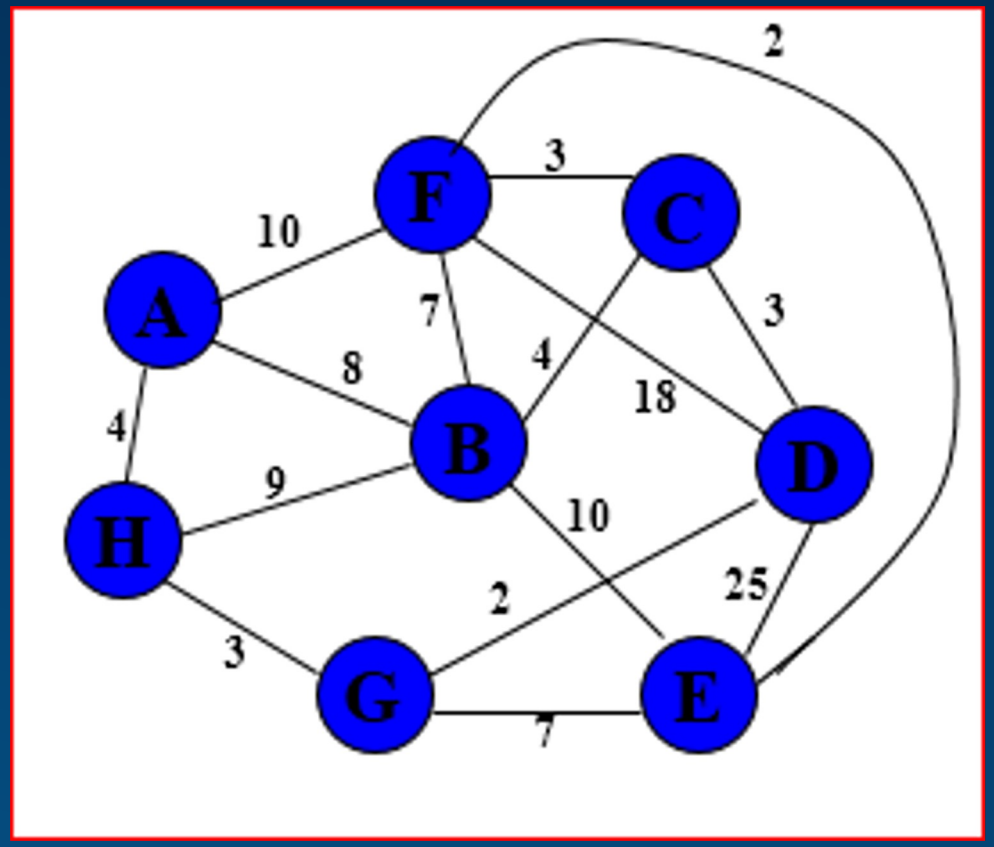
1. Initialize the spanning tree T to contain a vertex v1.
2. Choose an edge e = (v1, v2) of G such that v2 not equal to v1 and e has smallest weight among the edges of G incident with v1.
3. Select an edge e = (v2, v3) of G such that v2 is not equal to v3 and e has smallest weight among the edge of G incident with v2.
4. Suppose the edge e1, e2, e3, ... ei Then select an edge ei + 1 = (Vj, Vk) such that

(a) Vj ∈ {v1, v2, v3, ... vi, vi + 1} and

(b) Vk ∉ { v1, v2, v3, ... vi, vi + 1} such that ei+1 has smallest weight among the edge of G

5. Repeat the step 4 until (n – 1) edges have been chosen
6. Exit

Prim's Algorithm



Vertices	K	Edge	Path
A			
B			
C			
D			
E			
F			
G			
H			
MST			

x x x x x x x x x

x x x x x x x x x