Design and Analysis of Algorithm

Lecture-11:
Greedy Algorithm

Contents

Time Complexity

- The main time taking step is the sorting of all items in decreasing order of their value/weight ratio.
- If the items are already arranged in the required order, then while loop takes $O(n)$ time.
- The average time complexity of Quick Sort is $O(\text{n log n})$.
- Therefore, total time taken including the sort is $O(\text{n log n})$.

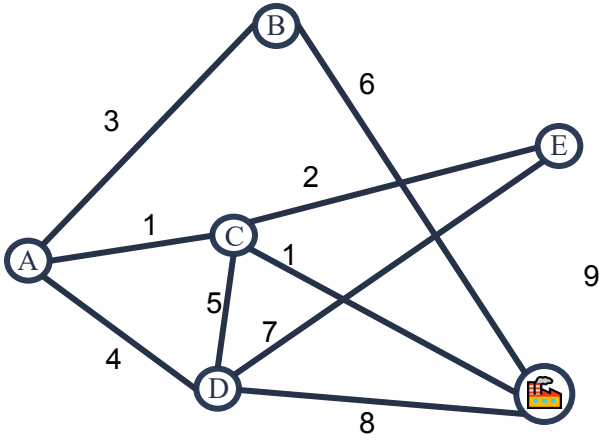
Knapsack Problem

Lemma 1. In case the sum of all the weights is $< m$, then $x_i = 1, 1 \leq i \leq n$ is an optimal solution

Lemma 2. All optimal solutions will fill the knapsack exactly

Minimum Spanning Tree

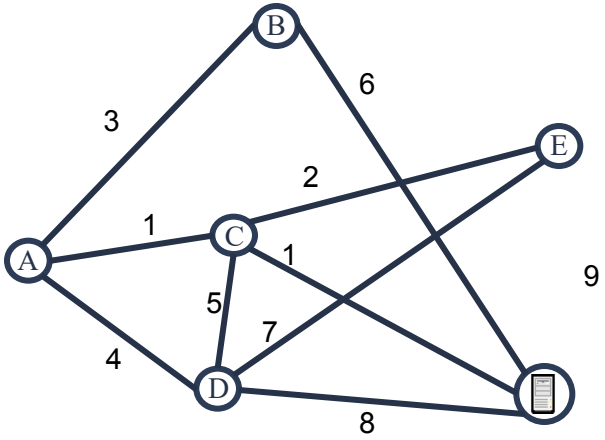
- Your friend at the electric company needs to choose where to build wires to connect all these cities to the plant.



She knows how much it would cost to lay electric wires between any pair of locations, and wants the cheapest way to make sure electricity from the plant to every city.

Minimum Spanning Tree

- Your friend at the ISP needs to choose where to build wires to connect all these cities to the Internet with fiber optic cable



She knows how much it would cost to lay cable wires between any pair of locations, and wants the cheapest way to make sure that every one is connected to server.

Minimum Spanning Tree

What do we need? A set of edges such that:

- Every vertex touches at least one of the edges. (the edges span the graph)
- The graph on just those edges is connected.
- The minimum weight set of edges that meet those conditions.

Constraint: It should not form cycle

Minimum Spanning Tree

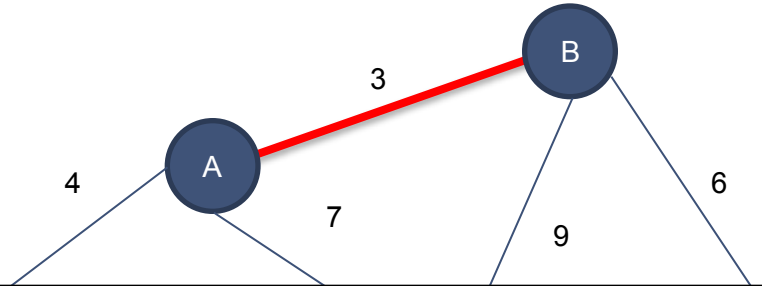
Given (connected) graph $G(V, E)$ a spanning tree T(V, E'):

- Is a sub-graph of G; that is, $E' \subseteq E$
- Forms a tree (no cycle)
- So, E' has |V| -1 edges

Edges are weighted: find minimum cost spanning tree

Strategy for construction:

- Add an edge of minimum cost that does not create a cycle (greedy algorithm)
- Repeat |V| -1 times



Two Algorithms

Prim: (build tree incrementally)

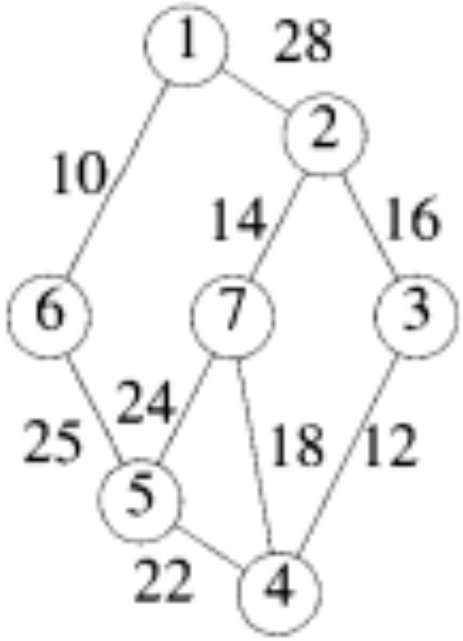
- Pick lower cost edge connected to known (incomplete) spanning tree that does not create a cycle and expand to include it in the tree

Kruskal: (build forest that will finish as a tree)

- Pick lowest cost edge not yet in a tree that does not create a cycle. Then expand the set of included edges to include it. (It will be somewhere in the forest.)

Algorithm

```
Algorithm Prim(E, cost, n, t)
// E is the set of edges in G. cost[1 : n, 1 : n] is the cost
// adjacency matrix of an n vertex graph such that cost[i, j] is
// either a positive real number or ∞ if no edge (i, j) exists.
// A minimum spanning tree is computed and stored as a set of
// edges in the array t[1 : n − 1, 1 : 2]. (t[i, 1], t[i, 2]) is an edge in
// the minimum-cost spanning tree. The final cost is returned.
{
    Let (k, l) be an edge of minimum cost in E;
    mincost := cost[k, l];
    t[1, 1] := k; t[1, 2] := l;
    for i := 1 to n do  // Initialize near.
        if (cost[i, l] < cost[i, k]) then near[i] := l;
        else near[i] := k;
    near[k] := near[l] := 0;
    for i := 2 to n − 1 do
    { // Find n − 2 additional edges for t.
        Let j be an index such that near[j] ≠ 0 and
        cost[j, near[j]] is minimum;
        t[i, 1] := j; t[i, 2] := near[j];
        mincost := mincost + cost[j, near[j]];
        near[j] := 0;
        for k := 1 to n do // Update near[ ].
            if ((near[k] ≠ 0) and (cost[k, near[k]] > cost[k, j]))
                then near[k] := j;
    }
    return mincost;
}
```



The time required by algorithm Prim is $O(n^2)$, where n is the number of vertices

Prim's algorithm

Starting from empty T (tree), choose a vertex at random and initialize

$$V = \{1\}, E' = \{\}$$

Choose the vertex u not in V such that edge weight from u to a vertex in V is minimal (greedy!)

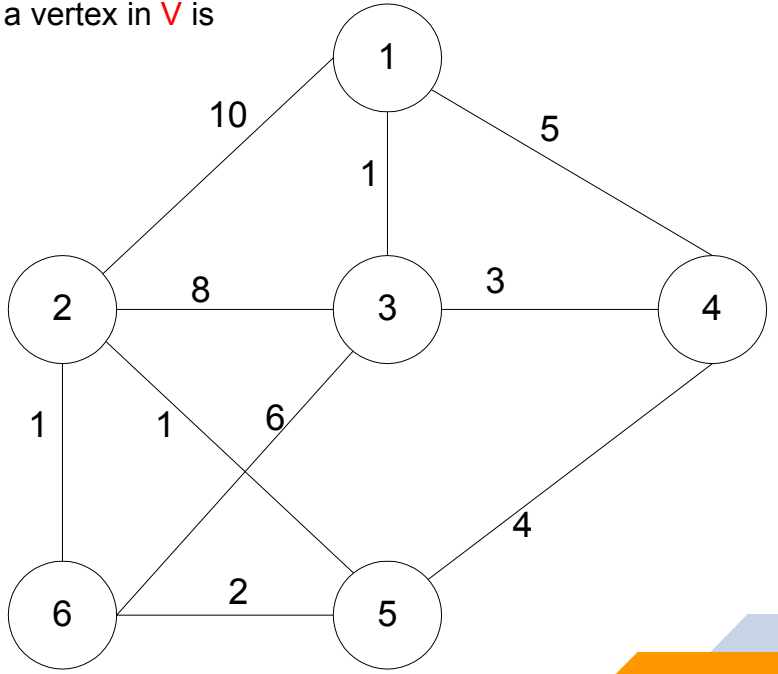
$V = \{1,3\}\ E' = \{(1,3)\}$

$V = \{1,3,4\}\ E' = \{(1,3),(3,4)\}$

$V = \{1,3,4,5\}\ E' = \{(1,3),(3,4),(4,5)\}$

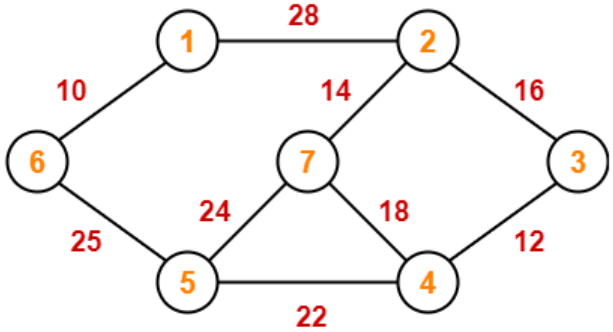
$V = \{1,3,4,5,2\}\ E' = \{(1,3),(3,4),(4,5),(5,2)\}$

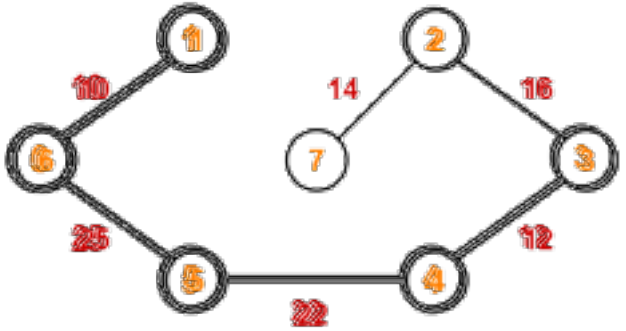
$V = \{1,3,4,5,2,6\}\ E' = \{(1,3),(3,4),(4,5),(5,2),(2,6)\}$



Questions

Construct the minimum spanning tree (MST) for the given graph using Prim's Algorithm-





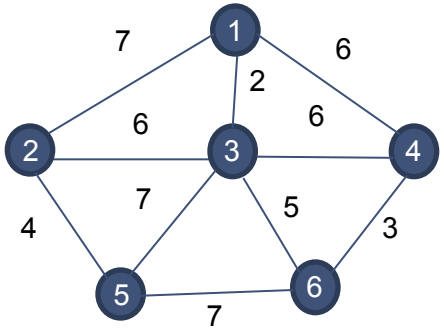
Total cost of Minimum Spanning Tree

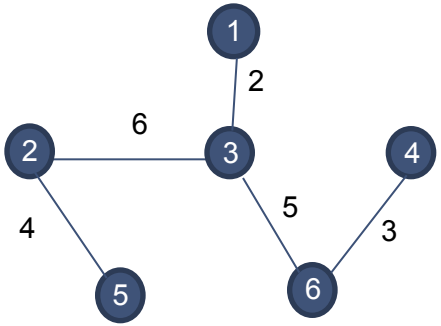
$= 10 + 25 + 22 + 12 + 16 + 14$

$= 99\ units$

Question

Apply Prim's algorithm to find MST of the following graph. What is the sequence of processing the nodes.

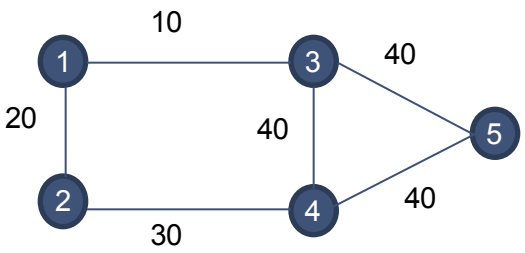


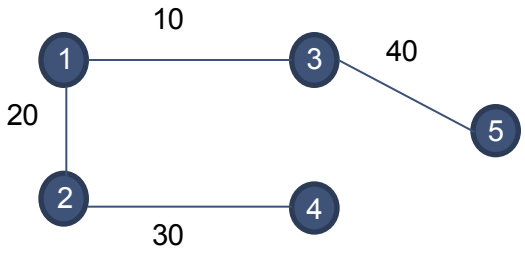


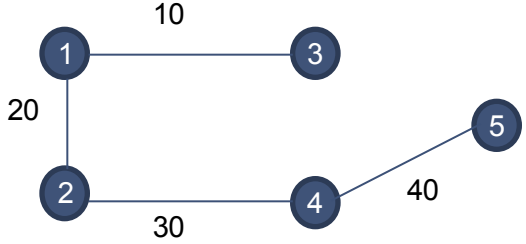
1 3 6 4 2 5

Question

Find the no. of minimum cost spanning tree of the following graph.







For the given graph more than one spanning tree is possible but minimum cost will remain same

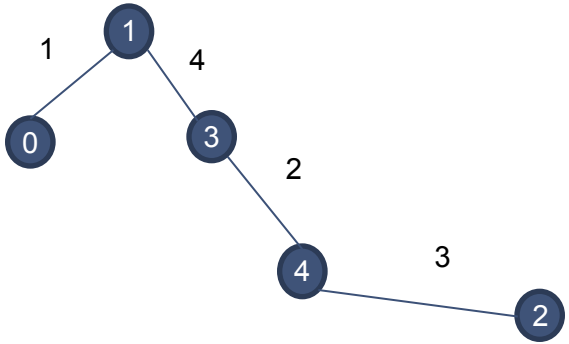
If the given graph contains more than one minimum cost spanning tree then in the given graph some edge weights are repeated.

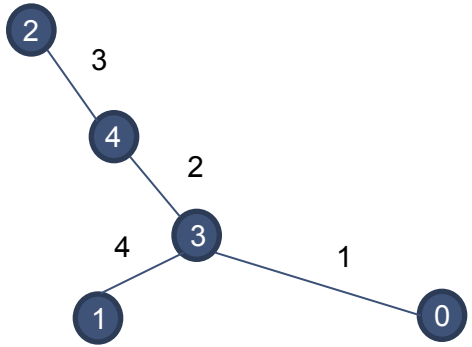
Question

Consider the complete undirected graph with vertex set $V = (0,1,2,3,4)$. The weight matrix of the edge (i,j) is given by

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{bmatrix} \end{array}$$

What is the cost of MST such that 0 is the leaf node





Question

$Q\ 1 \rightarrow$ Let $G(E, V)$ be undirected connected graph with distinct edge weight such that |E|>|V|. Let e_{max} be the edge with maximum weight and e_{min} be the edge with minimum weight then which of the following is false.

a) Every MST must contain $e_{\min}$
b) If $e_{\max}$ is in MST then it's removal must disconnect G
c) No MST contain $e_{\max}$
d) G has one unique MST.