NAME- RAM CHANDRA JANGIR
Subject- ADSA Programing Assignment
ROLL NO. - CS21M517

---

① Implement Prim's and Kruskal algorithm ?

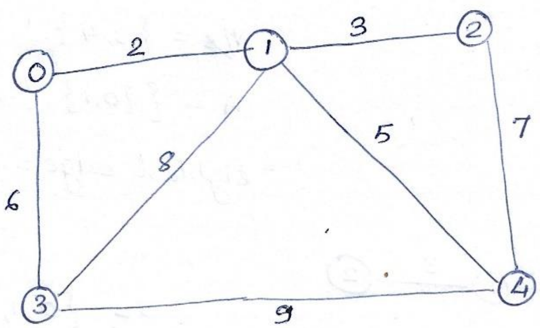
Prim's algorithm is as following:

step-1 Remove loops and parallel edges.
Put the edge with minimum weight in to spanning Tree.

step-2.
Continue the same process of adding edges that are incident to an existing vertex, never forming a simple circuit with those edges already in the tree.

step-3
Once (n-1) edges are added, stop the process.

Example



Source vertex : 0

Adjacency matrix for above graph =

	0	1	2	3	4
0	0	2	0	6	0
1	2	0	3	8	5
2	0	3	0	0	7
3	6	8	0	0	9
4	0	5	7	9	0

step-1

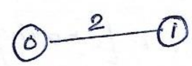
set $S = \{0\}$

Remaining vertices in the graph $V/S = \{1, 2, 3, 4\}$

$A = \{\}$

lightest edge $= \{0, 1\}$

step-2



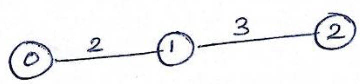
$S = \{0, 1\}$

$V/S = \{2, 3, 4\}$

$A = \{\{0,1\}\}$

lightest edge $= \{1, 2\}$

step-3



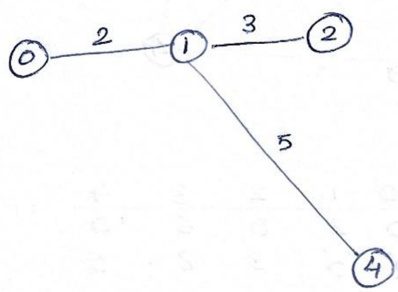
$S = \{0, 1, 2\}$

$V/S = \{3, 4\}$

$A = \{\{0,1\}, \{1,2\}\}$

lightest edge $= \{1, 4\}$

step-4

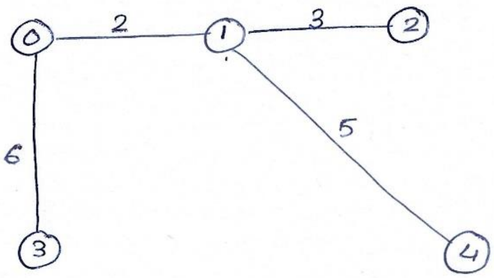


$S = \{0, 1, 2, 4\}$

$V/S = \{3\}$

$A = \{\{0,1\}, \{1,2\}, \{1,4\}\}$

lightest edge $= \{0, 3\}$



S = {0, 1, 2, 4, 3}

V-S = { }

A = { {0,1}, {1,2}, {1,4}, {0,3} }

lightest edge = { }.

so finally we get above minimum spanning tree with minimum cost = 2 + 3 + 5 + 6

= 16

Time complexity :

The time complexity of the Prim's algorithm is O((V+E) log V) because each vertex is inserted in the priority queue only once and insertion in priority queue takes logarithmic time.

① Kruskal Algorithm:

The kruskal's algorithm uses greedy approach for finding MST. It treats each of the node as an independent tree and connects with one another if any of them has lowest cost compared to all the other options that are available.

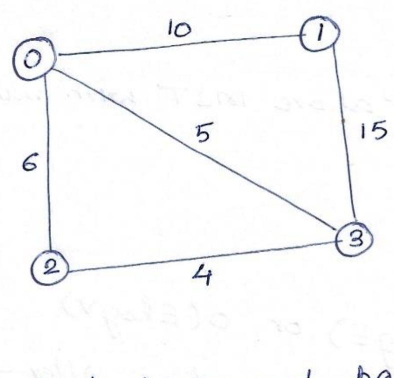
The Kruskal's algorithm follows below steps:

step-1
Removal of loops and parallel edges.

step-2 Arrangement of all edges in asscending order of cost.

step-3
Add the edge with least weight.

Example :



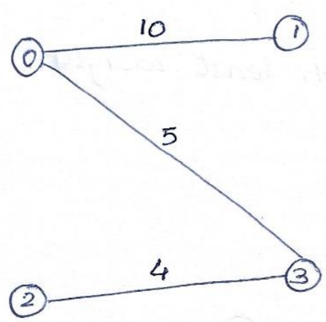
step-1 Removal of loops and parallel edges.

step-2

Arrangement of all edges in Ascending order.

After sorting

Weight	Source Vertex	Destination Vertex
4	2	3
5	0	3
6	0	2
10	0	1
15	1	3



So finally we get above MST with minimum cost

$= 4 + 5 + 10$

$= 19$

Time complexity

$= O(E\log E)$ or $O(E\log V)$

Sorting of edges takes $O(E\log E)$ time. After sorting we iterate through all edges and apply the find-union algorithm.

The find and union operations can take atmost $O(\log V)$ time.

Therefore, the overall time complexity is $O(E \log E)$ or $O(E \log V)$.