NAME: SNEHA MAGANAHALLI RAJENDRANATH

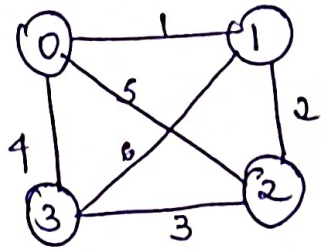
ROLL NUM: CS21M522

SUB : Data Structures

Lab Assignment - 01

(1) Implement prims and kruskal Algorithm

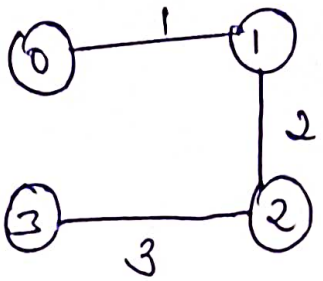
(a) prims Algorithm :-



adjacency Matrix

	0	1	2	3
0	0	1	5	4
1	1	0	2	6
2	5	2	0	3
3	4	6	3	0

MST output



Algorithm :

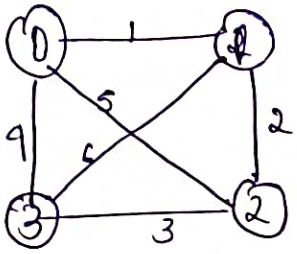
How does Prim's work???

(a) choose the smallest edge. (smallest weighted edge)

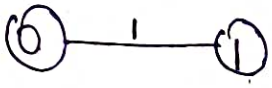
(b) choose the next smallest-weighted edge, which is connected to the already selected edge in step-(a)

(c) Repeat step (b) until you get MST.

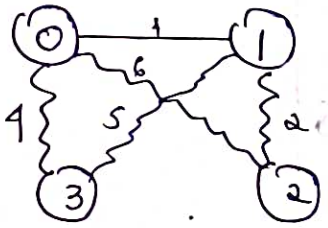
Tracing :



Select :



Compare :

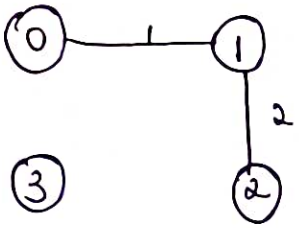


⇒ compare

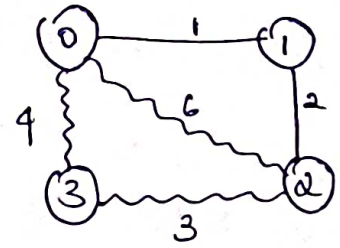
(2 → selected < 4 < 5 < 6)

compare among all the edges which are connected to the selected edge. Select the smallest-weighted edge.

Select :

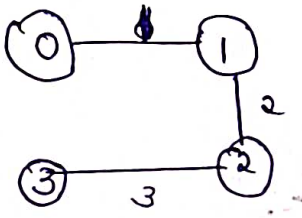


compare :



Select (2,3) ∵ $3 < 4 < 6$

select :



MST is formed !!!

Total sum of MST = $1+2+3 = 6$

Time complexity :-

Since we used the adjacency list. & Adjacency list is of $n \times n$ matrix, where $n \rightarrow$ no of vertices.

∴ prims algorithm has $O(n^2)$ time complexity.

$O(n^2)$

It can be reduced to $O(E \log V)$ by use of adjacency list instead of adjacency matrix. It also uses Min heap.

(b) Kruskal Algorithm :-

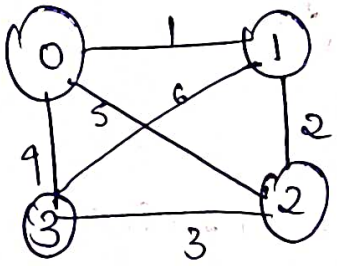
Steps:

(i) Find the minimum edge. Add to the result.

(ii) Find the next minimum-edge such that no cycles is formed. Add it to the result.

(iii) repeat-step (ii) untill you get a MST

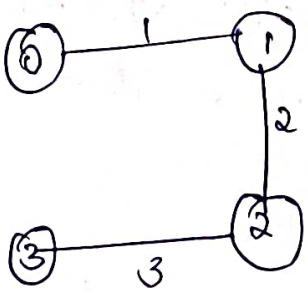
Tracing :



adjacency Matrix :

	0	1	2	3
0	0	1	5	4
1	1	0	2	6
2	5	2	0	3
3	4	6	3	0

MST output

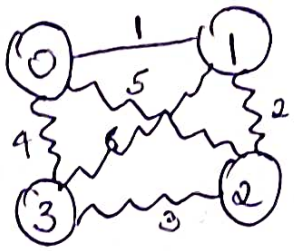


MST cost : 6

Tracing :

Select :

Compare :

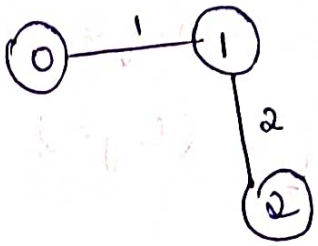


2 < 3 < 4 < 5 < 6

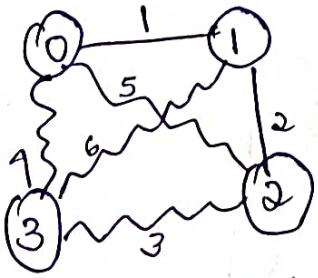
↓

Selected ∴ loop is also not formed

Select :



Compare :

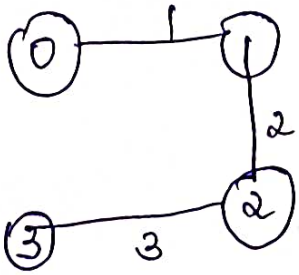


3 < 4 < 5 < 6

↓

Selected ∴ loop is also not formed.

Select :



∴ MST is formed.

∴ MST cost = 6.

Time complexity :-

$$\left[\begin{matrix}\text{Total time}\\ \text{complexity}\end{matrix}\right] = \left[\begin{matrix}\text{Sort all the}\\ \text{edges}\end{matrix}\right] + \left[\begin{matrix}\text{Find \& union on}\\ \text{each edge to}\\ \text{detect cycle}\end{matrix}\right]$$

If E = No of edges

V = no of vertices

Max no of edges in a graph = $^nC_2 = \frac{n(n-1)}{2} \approx n^2$

$$E_{max} = V^2$$

Apply log on both sides

$$\log E = 2 \log V$$

$$\boxed{O(\log E) \approx O(\log V)}$$

$$\left[\begin{matrix}\text{Total time}\\ \text{complexity}\end{matrix}\right] = [O(E \log E)] + [O(E \log E)]$$

(E: no of edges; $\log E$: find & union operation time)

$$= O(2E \log E)$$

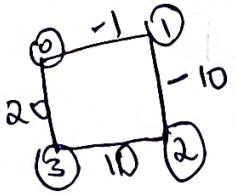
$$= O(E \log E)$$

Differences & similarity between prims & kruskal algorithm

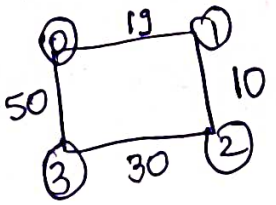
prims	Kruskal
○ MST algorithm where $E = \|V\| - 1$	○ MST algorithm
○ Time complexity $O(V^2)$ Improved $O(E \log E)$	$O(E \log E)$
○ works only on connected Graph.	○ works on disconnected components also.
○ Starts from any vertex.	○ starts from smallest weighted edge.
○ Both When negative weights are present it will not work. →	same applies.

Solution:-

○ Take maximum number as a constant. And add this constant to all the edges



⇒ Add +20 to all edges



] ⇒ all edges are positive