The Prim – Jarnik's Algo:-

1. Start with a single cluster C having some "root" vertex v. $C = \{v\}$

2. Choose a minimum-weight edge $e = \{u, v\}$ connecting v in the cloud C and u outside the cloud C

3. Add vertex u to the cloud $C = \{u, v\}$

4. Repeat until a MST is formed.

For efficiency we have a label $D[u]$ for each vertex u outside the cloud C.

- stores the weight of the best current edge for joining u to the cloud C. [the smallest wt of an edge connecting u to a vertex in cloud]

∴ this reduces the no/: of edges we need to consider to decide which vertex should join the cloud next.

(1)

Algorithm PrimJarnikMST(G):

Input: A weighted connected graph G with n vertices and m edges.

Output: A minimum spanning tree T for G.

Pick any vertex $u \in V$ do

$D[u] \leftarrow 0$

for each vertex $u \neq V$ do

$D[u] \leftarrow \infty$.

Initialize $T \leftarrow \emptyset$.

Initialize a priority queue Q with an item $((u, null), D[u])$ for each vertex u, where $(u, null)$ is the element and $D[u]$ is the key.

while Q is not empty do

$(u, e) \leftarrow Q.removeMin()$

Add vertex u and edge e to T.

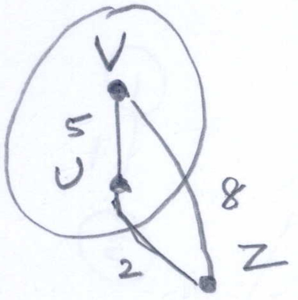
for each vertex z adjacent to u such that z is in Q do if $w(u,z) < D[z]$ then

$D[z] \leftarrow w(u,z)$

Change the element of vertex z in Q to

$(z, (u,z))$

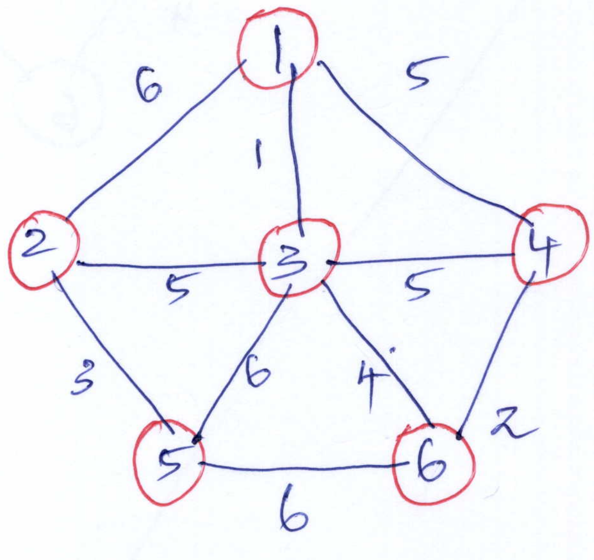
change the key of vertex z in Q to $D[z]$



Here $D[z]$ is changed to 7.

return the tree T.

Running Time :- $O(m \log n)$



Source 1.

$D[1] = 0$

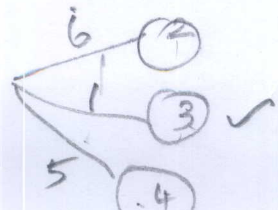
V	D[v]
1	0
2	∞
3	∞
4	∞
5	∞
6	∞

$T = \phi$

Step1:

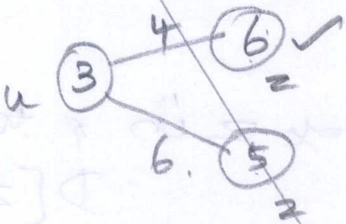
$T = \{1\}$

$u = 1$

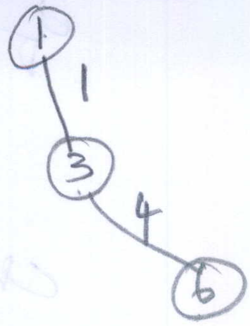


$u = 3$

$T = \{1, 3\}$

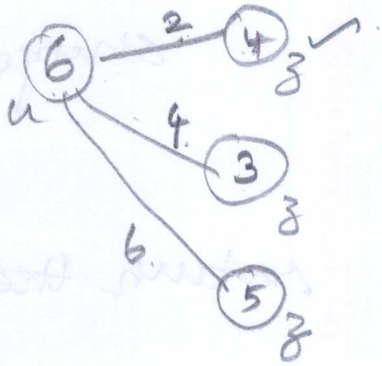


V	D[v]	
1	0	
~~3~~	~~1~~ ✓	
2	6	
4	5	
5	$\infty \longrightarrow 6.$	
6	$\infty \xrightarrow{\checkmark} 4$	min

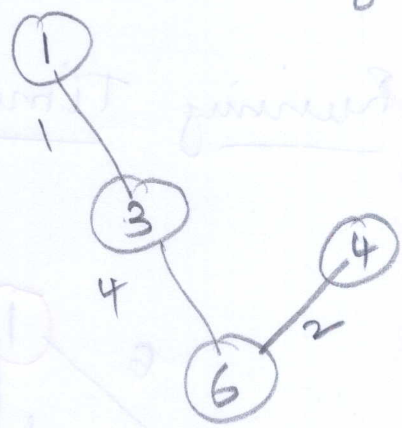


$u = 6$

$T = \{1, 3, 6\}$



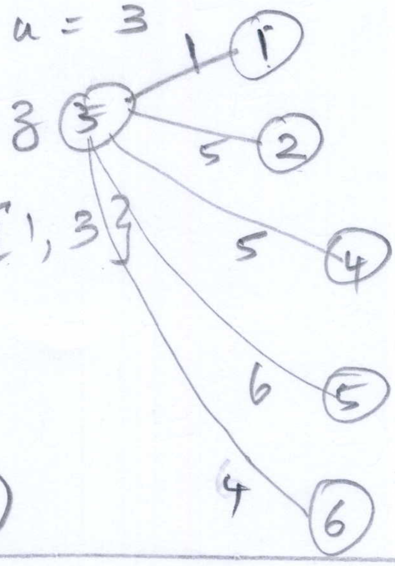
V	D[v]	
1	0	
2	6	
3	1	no change
4	2	change
5	6	no change.



V	D[v]
1	0
2	6
3	1
4	5
5	∞
6	∞

STEP 2:

min = u = 3



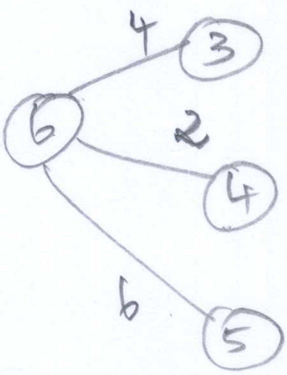
T = {1, 3}

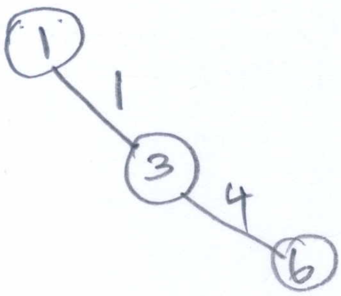
V	D[v]	
1	0	×
2	~~6~~	5
3	1	
4	5	×
5	~~∞~~	6
6	~~∞~~	4

STEP 3:

min = u = 6

T = {1, 3, 6}



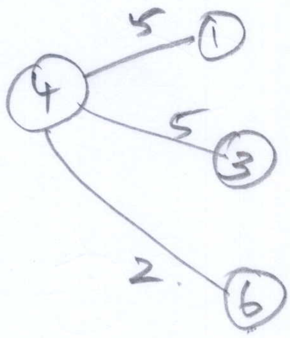


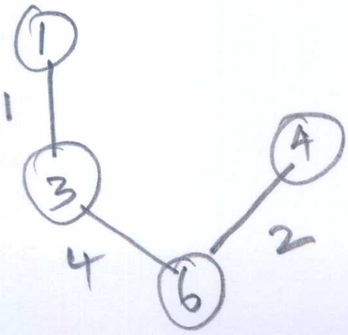
V	old D[v]		New
~~1~~	~~0~~		0
2	5		5
~~3~~	~~1~~	×	1
4	~~5~~	2	2
5	6	×	6
~~6~~	~~4~~		4

STEP 4:

min = u = 4

T = {1, 3, 6, 4}



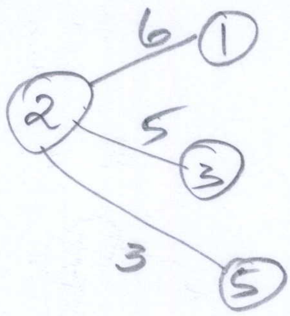


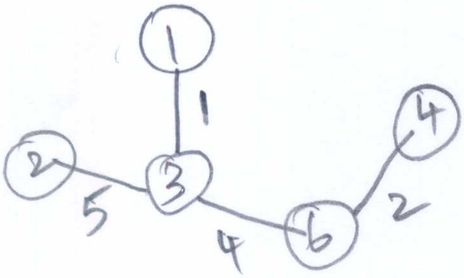
V	old D[v]	New	
~~1~~	~~0~~	~~×~~	
2	5	5	(edge 3)
~~3~~	~~1~~	~~×~~	
~~4~~	~~2~~	~~2~~	
5	6	6	
~~6~~	~~4~~	~~2~~	

STEP 5:

$min = u = 2$

$T = \{1, 3, 6, 4, 2\}$





V	old D[V]	New
~~1~~	~~0~~	~~X~~
~~2~~	~~5~~	~~5~~
~~3~~	~~1~~	~~x~~
~~4~~	~~2~~	~~2~~
~~5~~	~~6~~	3 — {with edge 2}
~~6~~	~~4~~	~~4~~

∴ STEP 6:

$T = \{1, 3, 6, 4, 2, 5\}$

∴ MST =