Prim's Algorithm

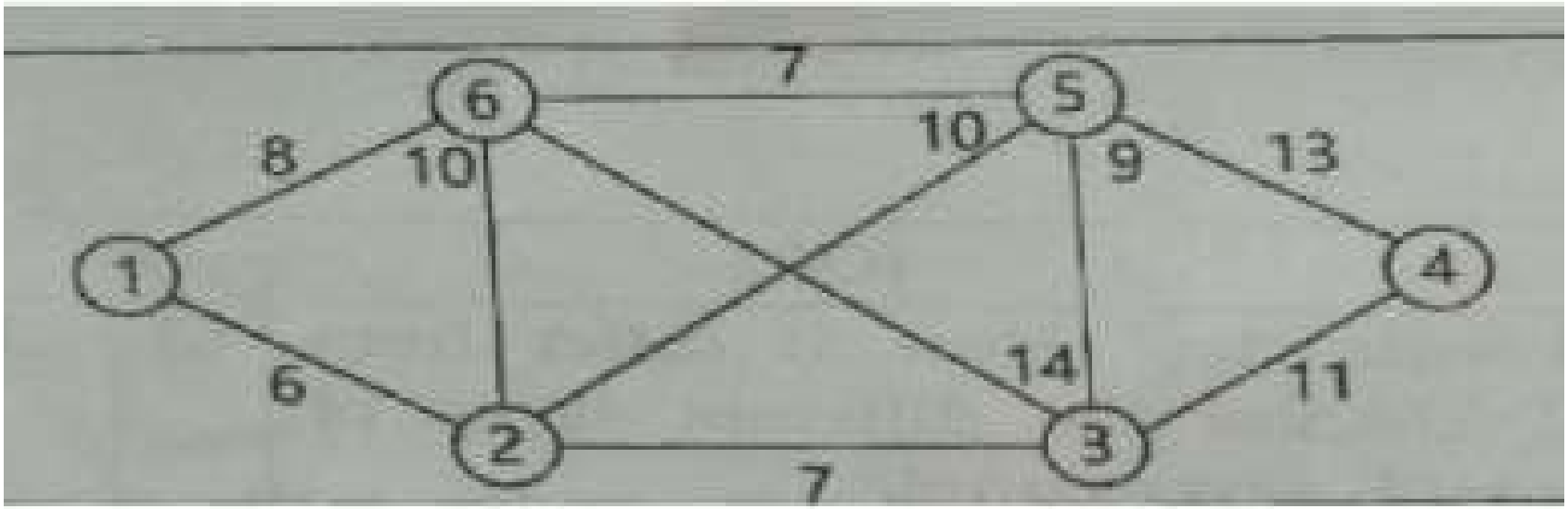
New vertex add to the tree by choosing the edge (u,v) such that the cost of (u,v) is the smallest among all edges where u is in the tree and v is not

Vertex	*Known*	d_v	p_v
A			
B			
C			
D			
E			
F			
G			
H			

Known – T / F

d_v ***- Weight of the shortest edge connecting v to a known vertex***

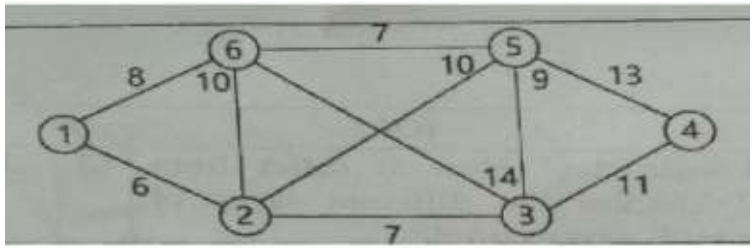
p_v ***- Last vertex to cause a change in*** d_v


7
6
5
8
10
10
9
13
1
4
6
14
11
2
3
7

Step-1

Initialize Configuration

V	*K*	d_v	p_v
1	F	∞	–
2	F	∞	–
3	F	∞	–
4	F	∞	–
5	F	∞	–
6	F	∞	–

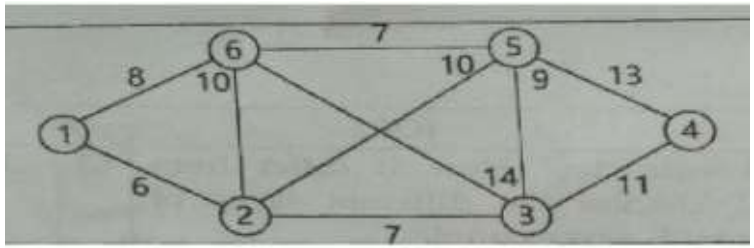


Step-2

Start with any node, say 1

After 1 is declared known

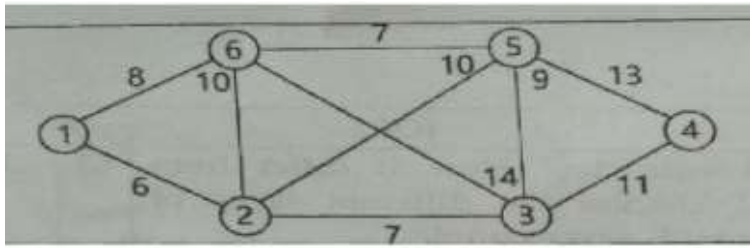
V	*K*	d_v	p_v
1	T	0	–
2	F	6	1
3	F	∞	–
4	F	∞	–
5	F	∞	–
6	F	8	1



Step-3

After 2 is declared known

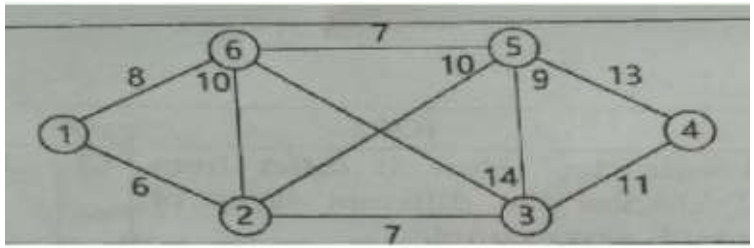
V	*K*	d_v	p_v
1	T	0	–
2	T	6	1
3	F	7	2
4	F	∞	–
5	F	10	2
6	F	8	1



Step-4

After 3 is declared known

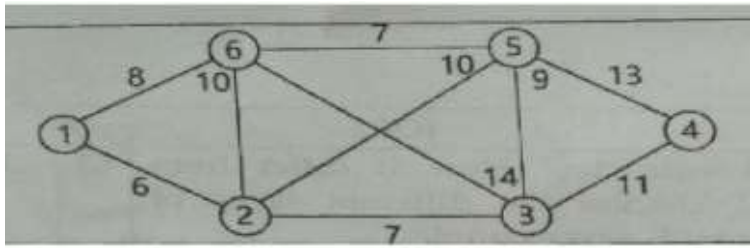
V	*K*	d_v	p_v
1	T	0	–
2	T	6	1
3	T	7	2
4	F	11	3
5	F	9	3
6	F	8	1



Step-5

After 6 is declared known

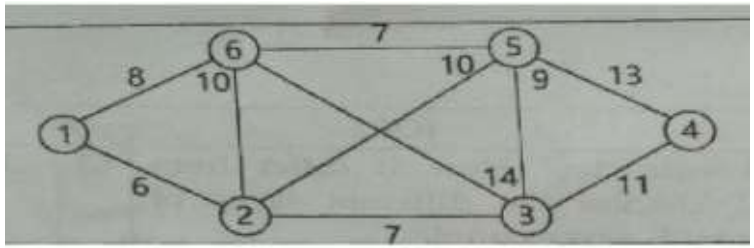
V	*K*	d_v	p_v
1	T	0	–
2	T	6	1
3	T	7	2
4	F	11	3
5	F	7	6
6	T	8	1

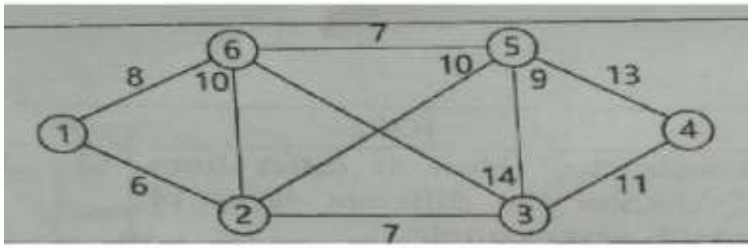


Step-6

After 5 is declared known

V	*K*	d_v	p_v
1	T	0	–
2	T	6	1
3	T	7	2
4	F	11	3
5	T	7	6
6	T	8	1





Step-6

After 4 is declared known

V	*K*	d_v	p_v
1	T	0	–
2	T	6	1
3	T	7	2
4	T	11	3
5	T	7	6
6	T	8	1

Done

Cost of Minimum
Spanning Tree = $\Sigma\, d_v$ = **39**