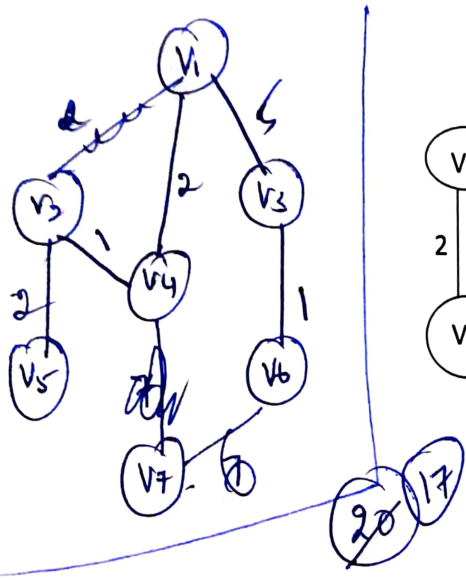
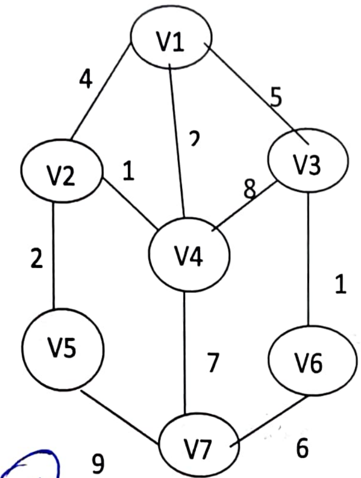
- ***Prim's Algorithm***

According to prim's algorithm, a minimum spanning tree grows in successive stages. The prim's algorithm find a new vertex to add it to the tree by choosing the edge **<v_i,v_j>**,the smallest among all edges, where **v_i** I the tree and **v_j** is yet to be included in the tree.The prim's algorithm can easily be implemented using the adjacency matrix representation of a graph. Let us now illustrate the above method of finding a minimum spanning tree





	V1	V2	V3	V4	V5	V6	V7
V1	0	4	5	2	∞	∞	∞
V2	4	0	∞	1	2	∞	∞
V3	5	∞	0	8	∞	1	∞
V4	2	1	8	0	∞	∞	7
V5	∞	2	∞	∞	0	∞	9
V6	∞	∞	1	∞	∞	0	6
V7	∞	∞	∞	7	9	6	0

Figure 4.22 Graph

n log n

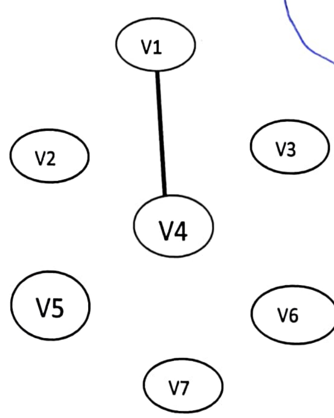
edges E log E

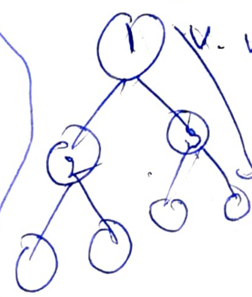
n lo

We start with v1 and pick the smallest entry; thus v4 is the nearest neighbour to v1

Step 1:

V1	0	T	0
V2	4	F	V1
V3	5	F	V1
V4	2	F	V1
V5	-	F	-
V6	-	F	-
V7	-	F	-





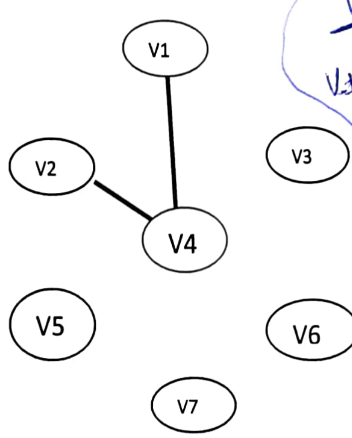
V log V

V-1=E

V=E+1

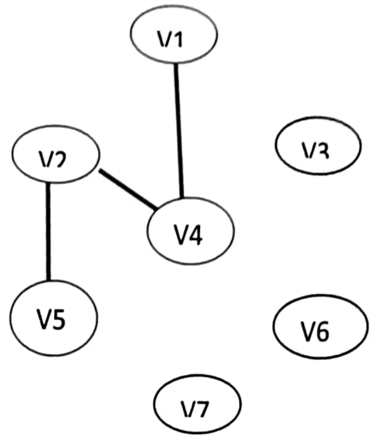
Step 2:

V1	0	T	0
V2	1	F	V4
V3	5	F	V1
V4	2	T	V1
V5	-	F	-
V6	-	F	-
V7	7	F	V4



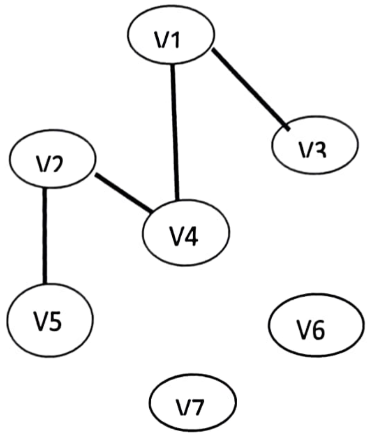
Step 3:

V1	0	T	0
V2	1	T	V4
V3	5	F	V1
V4	2	T	V1
V5	2	F	V2
V6	-	F	-
V7	7	F	V4



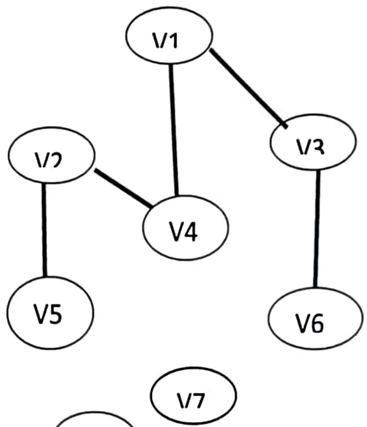
Step 4:

V1	0	T	0
V2	1	T	V4
V3	5	F	V1
V4	2	T	V1
V5	2	T	V2
V6	-	F	-
V7	7	F	V4



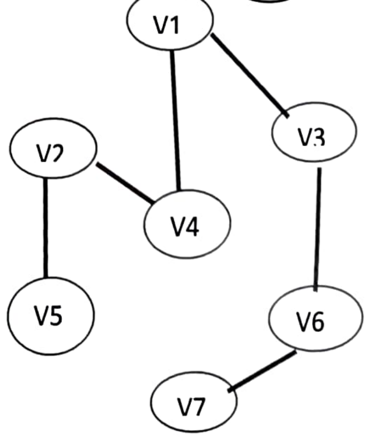
Step 5:

V1	0	T	0
V2	1	T	V4
V3	5	T	V1
V4	2	T	V1
V5	2	T	V2
V6	1	F	V3
V7	7	F	V4



Step 6:

V1	0	T	0
V2	1	T	V4
V3	5	T	V1
V4	2	T	V1
V5	2	T	V2
V6	1	T	V3
V7	6	F	V6



The minimum spanning tree obtained is

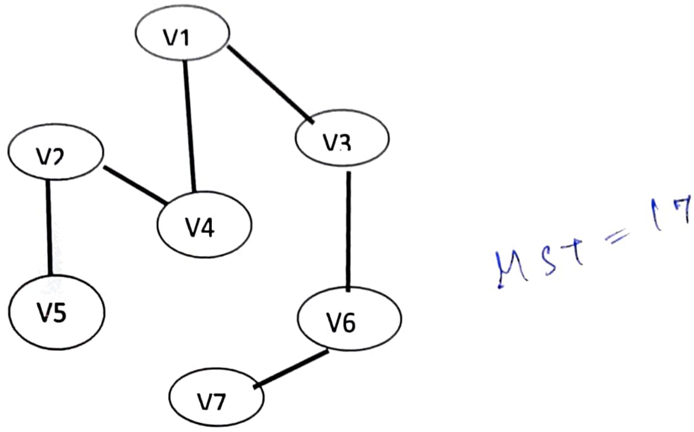


Figure 4.23 Minimum Spanning Tree

4.6 Shortest Path Problem

This problem of a graph is about finding a path between two vertices in such a way that this path will satisfy some criteria of optimization.for example,for a non-weighted graph,the number of edges will be minimum and for a weighted graph,the sum of weights on all edges in the path will be minimum.

4.6.1 All Pair Shortest Path

In the all-pair shortest problem we must find the shortest paths between all pairs of vertices, vi, vj.A few important algorithms are warshall's algorithm,folyd's Algorithm

Floyd's algorithm :

warshall's algorithm shows the presence or absence of any path between a pair of vertices.it does not take into account the weights of edges.

If weights are to be taken in account and if we are interested in the length of the shortest path between any pair of vertices,then another classical solution is known asFloyd's algorithm .

It use two function namely, Min(x,y) and combine(p1,p2).min(x,y) returns the minimum value between x and y,and combine(p1,p2) returns the concationation of two path p1 and p2 resulting in single path.

Step 1: Let G=<N,A> be a directed graph 'N' is a set of nodes and 'A' is the set of edges.

Step 2: Each edge has an associated non-negative length.

Step 3: We want to calculate the length of the shortest path between each pair of nodes.

Step 4: Suppose the nodes of G are numbered from 1 to n, so N={1,2,...N},and suppose G matrix L gives the length of each edge, with L(i,j)=0 for i=1,2...n,L(i,j)>=for all i& j, and L(i,j)=infinity, if the edge (i,j) does not exist.

Step 5: The principle of optimality applies: if k is the node on the shortest path from i to j then the part of the path from i to k and the part from k to j must also be optimal, that is shorter.

Step 6: First, create a cost adjacency matrix for the given graph.

The constraint is to use minimum number of colors. A map can be represented as a graph wherein a node represents a region and an edge between two regions denote that the two regions are adjacent.

The graphic representation of a map

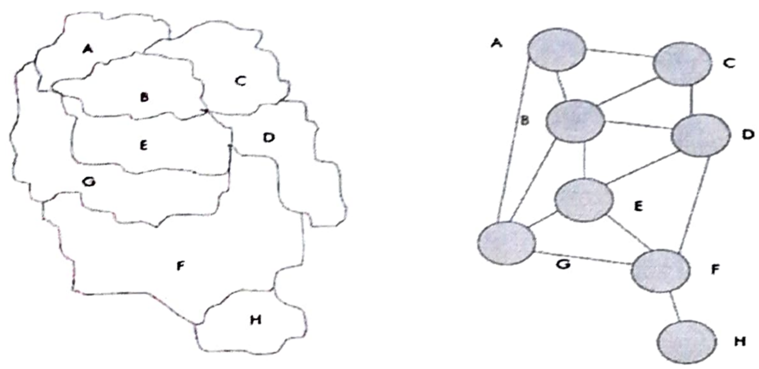


Figure 4.18 Coloring of map example

4.5 Minimum Spanning Trees

The spanning tree of a graph G can be defined as a tree which includes all the vertices of G.In graph traversal,we have seen that the DFS and BFS traversal result in two trees

DFS – spanning tree and BFS – spanning tree

The minimum spanning tree problem is related to the weighted graph, where we find a spanning tree so that the sum of all the weighted of all edges in the tree is minimum.

Two efficient method available for finding a spanning tree are

- Kruskal's algorithm
- Prim's algorithm

- ***Kruskal's algorithm***

To obtain a minimum spanning tree of a graph, a novel approach was devised by J.B KRUSKAL known as kruskal's algorithm.

Algorithm kruskal

1.list all the edges of the graph G in the increasing order of weights.

2.Select the smallest edge from the list and add it into the spanning tree, if the inclusion of this edge does not make a cycle

3.if the selected edge with smallest weight forms a cycle, remove it from the list.

4.repeat step 2-3 until the tree contains n-1 edges or list is empty

If the tree contains less than n-1 edges and the list is empty, no spanning tree is possible for the graph, else return minimum spanning tree

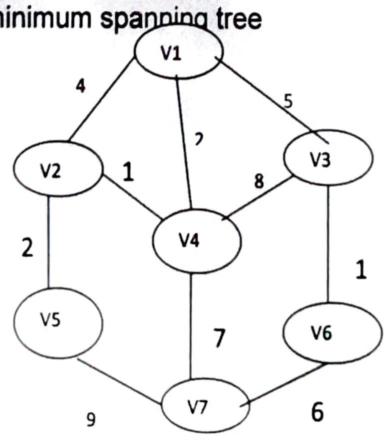


Figure 4.19 Graph

List of edges in non descending order

Edges	Weight	selection
V2-v4	1	
V3-v6	1	
V1-v4	2	
V2-v5	2	
V1-v2	4	
V1-v3	5	
V6-v7	6	
V4-v7	7	
V3-v4	8	
V5-v7	9	

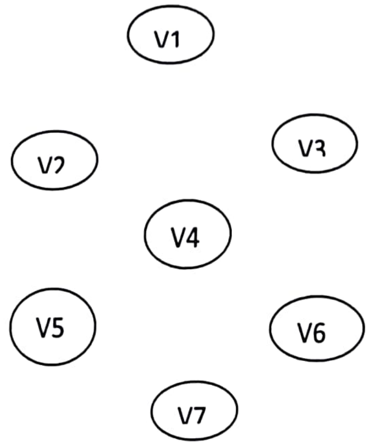
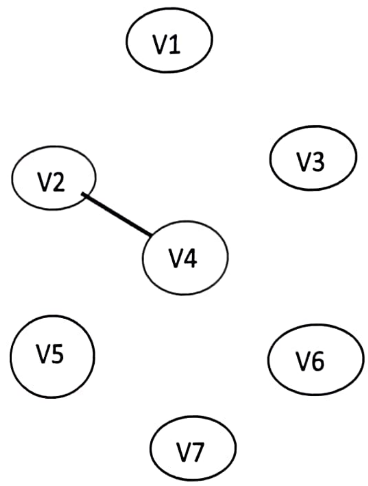


Figure 4.20 kruskal's algorithm example

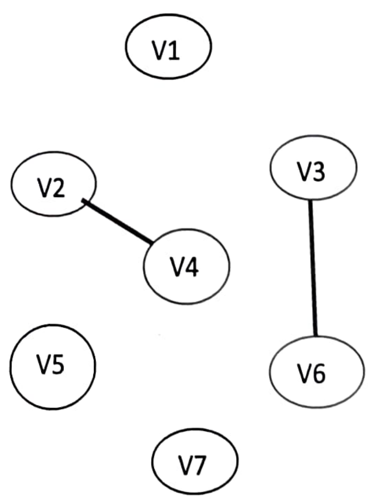
Step1:

Edges	Weight	selection
V2-v4	1	accept
V3-v6	1	
V1-v4	2	
V2-v5	2	
V1-v2	4	
V1-v3	5	
V6-v7	6	
V4-v7	7	
V3-v4	8	
V5-v7	9	



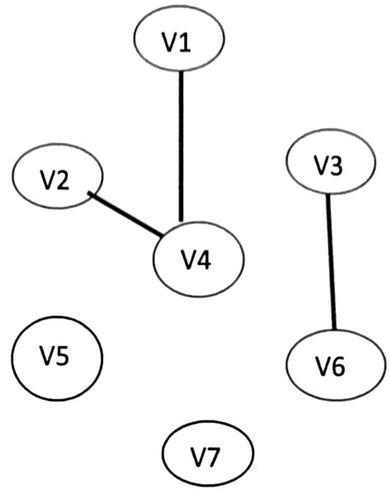
Step 2:

Edges	Weight	selection
V2-v4	1	accept
V3-v6	1	accept
V1-v4	2	
V2-v5	2	
V1-v2	4	
V1-v3	5	
V6-v7	6	
V4-v7	7	
V3-v4	8	
V5-v7	9	



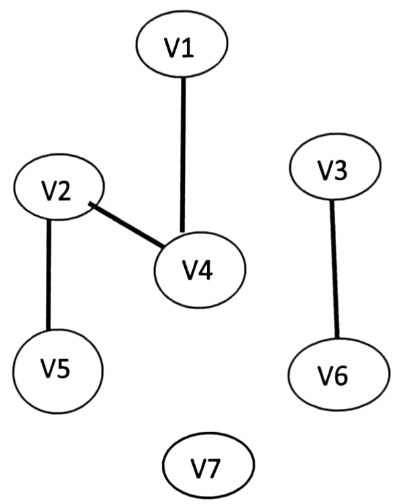
Step 3:

Edges	Weight	selection
V2-v4	1	accept
V3-v6	1	accept
V1-v4	2	accept
V2-v5	2	
V1-v2	4	
V1-v3	5	
V6-v7	6	
V4-v7	7	
V3-v4	8	
V5-v7	9	



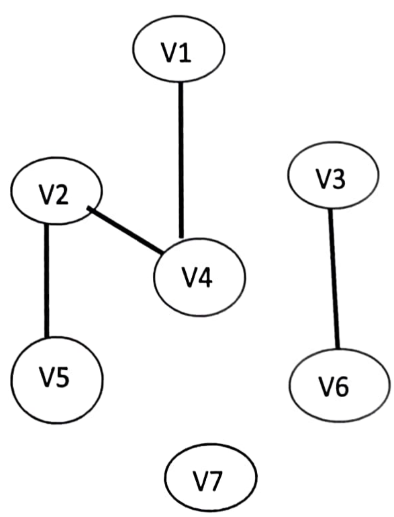
Step 4:

Edges	Weight	selection
V2-v4	1	accept
V3-v6	1	accept
V1-v4	2	accept
V2-v5	2	accept
V1-v2	4	
V1-v3	5	
V6-v7	6	
V4-v7	7	
V3-v4	8	
V5-v7	9	



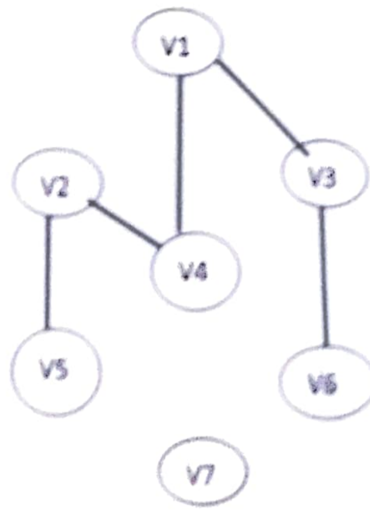
Step 5: acceptance of edge (v1-v2) create cycle ,so remove it from the list

Edges	Weight	selection
V2-v4	1	accept
V3-v6	1	accept
V1-v4	2	accept
V2-v5	2	accept
V1-v2	4	reject
V1-v3	5	
V6-v7	6	
V4-v7	7	
V3-v4	8	
V5-v7	9	



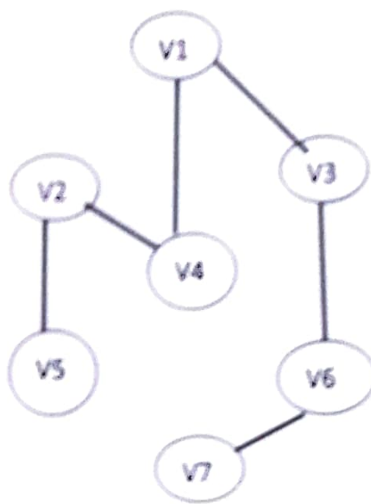
Step 6:

Edges	Weight	selection
V2-v4	1	accept
V3-v6	1	accept
V1-v4	2	accept
V2-v5	2	accept
V1-v2	4	reject
V1-v3	5	accept
V6-v7	6	
V4-v7	7	
V3-v4	8	
V5-v7	9	

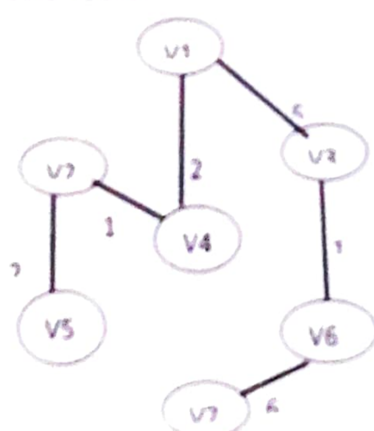


Step 7:

Edges	Weight	selection
V2-v4	1	accept
V3-v6	1	accept
V1-v4	2	accept
V2-v5	2	accept
V1-v2	4	reject
V1-v3	5	accept
V6-v7	6	accept
V4-v7	7	
V3-v4	8	
V5-v7	9	



The minimum spanning tree obtained is



MST = 17

Figure 4.21 Minimum Spanning Tree