Assignment No: B-11

Title: - To study and implement the construction of minimum spanning tree.

Index Terms: class, objects, MST, Prims, Cycle

Problem Statement: You have a business with several offices; you want to lease phone lines to connect them up with each other; and the phone company charges different amounts of money to connect different pairs of cities. You want a set of lines that connects all your offices with a minimum total cost. Solve the problem by suggesting appropriate data structures.

Theory:

Spannin• Tree: Any tree, which consists solely of edges in graph G and includes all the vertices in G is called as a spanning tree. Thus for a given connected graph there are multiple spanning trees possible. For a maximal connected graph having n vertices the number of different possible spanning trees is equal to (n!).

Minimum Spanning Tree: The minimum cost or minimum weightage spanning tree is called as Minimum Spanning Tree. To obtain a minimum spanning tree for a given connected graph, we can use Prim,s algorithm (vertex by vertex) or Kruskal‟ s algorithm (edge by edge).

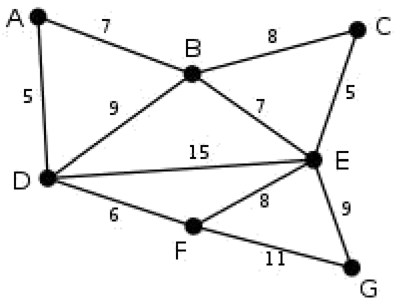
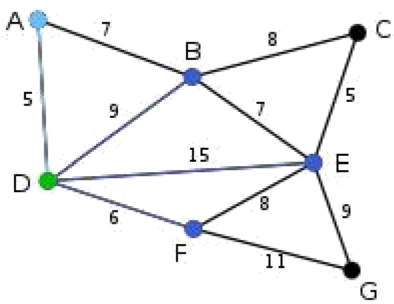
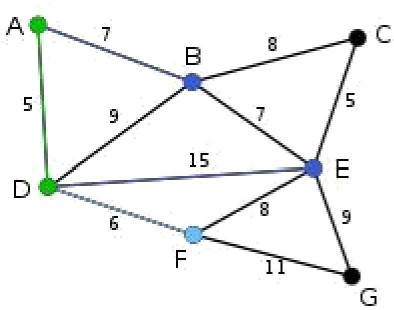
Prim's Algorithm: In computer science, Prim's algorithm is a greedy algorithm that finds a minimum spanning tree for a connected weighted undirected graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized.

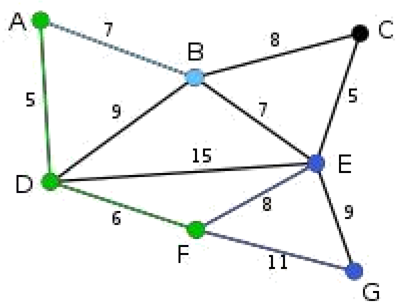
All vertices of a connected graph are included in minimum cost spanning tree. Prims algorithm starts from one vertex and grows the rest of tree by adding one vertex at a time by adding associated edge in set T. This algorithm builds a tree by iteratively adding edges until all vertices are visited. The resultant tree is a minimum spanning tree. At each iteration it **selects the vertex** and associated edge having minimum cost or weightage that does not create a cycle. The algorithm is:

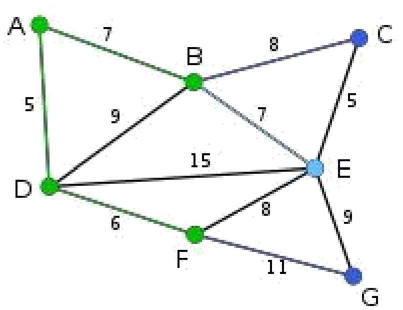
Applications of spanning trees:

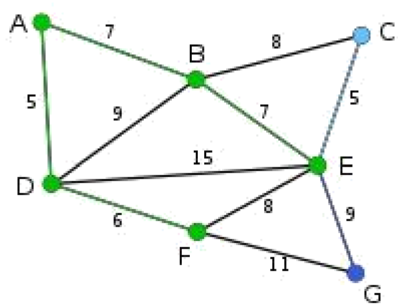
- To find independent set of circuit equations for an electrical network. By adding an edge from set B to spanning tree we get a cycle and then Kirchhoff‟ s second law is used on the resulting cycle to obtain a circuit equation. Thus the total number of independent circuit equation we get is equal to the number of edges in set B.
- Using the property of spanning trees we can select the spanning tree with (n-1) edges such that total cost is minimum if each edge in a graph represents cost (weightage). For example, a communication network between number of cities.

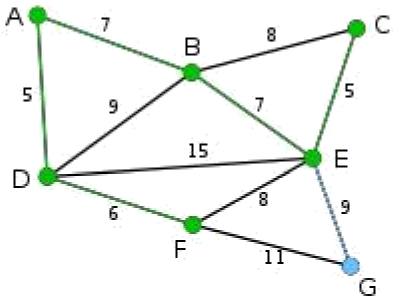
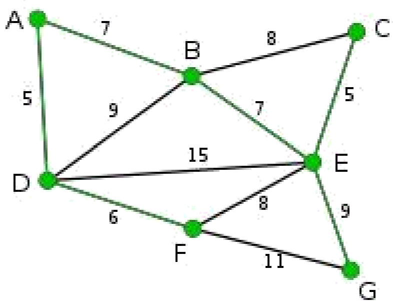
Example run

Image	U	Edge(u,v)	V \ U	Description
	{}		{A,B,C,D, E,F,G}	This is our original weighted graph. The numbers near the edges indicate their weight.
	{D}	(D,A) = 5 **V** (D,B) = 9 (D,E) = 15 (D,F) = 6	{A,B,C,E, F,G}	Vertex **D** has been arbitrarily chosen as a starting point. Vertices **A**, **B**, **E** and **F** are connected to **D** through a single edge. **A** is the vertex nearest to **D** and will be chosen as the second vertex along with the
	{A,D}	(D,B) = 9 (D,E) = 15 (D,F) = 6 **V**	{B,C,E,F, G}	The next vertex chosen is the vertex nearest to *either* **D** or **A**. **B** is 9 away from **D** and 7 away from **A**, **E** is 15, and **F** is 6. **F**

		(A,B) = 7		is the smallest distance away, so we highlight the vertex **F** and the arc **DF**.
	{A,D,F}	(D,B) = 9 (D,E) = 15 (A,B) = 7 **V** (F,E) = 8 (F,G) = 11	{B,C,E,G }	The algorithm carries on as above. Vertex **B**, which is 7 away from **A**, is highlighted.

	{A,B,D,F}	(B,C) = 8 (B,E) = 7 **V** (D,B) = 9 cycle (D,E) = 15 (F,E) = 8 (F,G) = 11	{C,E,G}	In this case, we can choose between **C**, **E**, and **G**. **C** is 8 away from **B**, **E** is 7 away from **B**, and **G** is 11 away from **F**. **E** is nearest, so we highlight the vertex **E** and the

	{A,B,D,E,F}	(B,C) = 8 (D,B) = 9 cycle (D,E) = 15 cycle (E,C) = 5 **V** (E,G) = 9 (F,E) = 8 cycle (F,G) = 11	{C,G}	Here, the only vertices available are **C** and **G**. **C** is 5 away from **E**, and **G** is 9 away from **E**. **C** is chosen, so it is highlighted along with the
	{A,B,C,D,E,F }	(B,C) = 8 cycle (D,B) = 9 cycle (D,E) = 15 cycle (E,G) = 9 **V** (F,E) = 8 cycle (F,G) = 11	{G}	Vertex **G** is the only remaining vertex. It is 11 away from **F**, and 9 away from **E**. **E** is nearer, so we highlight **G** and the arc **EG**.

 	{A,B,C,D,E,F, G}	(B,C) = 8 cycle (D,B) = 9 cycle (D,E) = 15 cycle (F,E) = 8 cycle (F,G) = 11 cycle	{}	Now all the vertices have been selected and the [minimum spanning tree] is shown in green. In this case, it has weight 39.

Class node is not necessary here… since we are implementing using adjacency Matrix.	class Graph
	{
	public:
	int cost[20][20], mincost, n; Graph()
	{
	for(i=1;i<=10;i++)
	for(j=1;j<=10;j++)
	{
	cost[i][j]=-1;
	}
	mincost=0;
	}
	void Create();
	void Display();
	void Prims();
	};

Algorithm Create ()

// This algorithm is used to read input undirected graph from user.

```
{
Read( no of nodes as n);
Repeat()
{
Write("Enter starting and ending vertices and its cost");
Read(v1 , v2 and c);
cost[v1][v2]=cost[v2][v1]=c;
Write("Do You Want To Enter More Edges ");
}until(false);
}
```

Algorithm Display ()

// This algorithm is used to print undirected graph given by user.

```
{
for( i=1 to n) do
for(j=1 to n ) do
Write(cost[i][j]);
}
```

Algorithm Prims()

// This algorithm is used to find MST and its cost using Prims Logic

```
{
for(i=1 to n) do
visit[i]=0;
```

```
Read ("Enter Staring Vertex in s");
Visit[s]=1;
for(k=1 to n-1) do
{
min=999;
for(i=1 to n) do {
for(j=1 to n) do
{
if(visit[i]==1 && visit[j]==0) // look for unvisited vertex from visited
{
if(cost[i][j]!=-1 && min>cost[i][j]) find near with min cost
{
min=cost[i][j];
row=i;
col=j;
}
}
}
}
Write(Selected Edge in MST is row and col);
mincost=mincost+min;
visit[col]=1;
cost[row][col]=-1;
cost[col][row]=-1;
}
Write(Total Min Cost as mincost);
}
```

Frequently Asked Questions:

1) What is a minimum spanning tree?
2) What is weighted graph?
3) What are the Prim's algorithms?
4) What are the applications of the minimal spanning tree?
5) How does Prim's algorithm work?
6) What is the Complexity of prim's algorithm?
7) What is the application of prim's algorithm?
8) State the advantages of the prim's algorithm?

Flowcharts:

Conclusions: