

Name : Lakhan Kumawat

Roll No : 1906055

Branch : Cse -1

Course : CSL4403

Program : To implement Prims Algorithm minimum weight spanning tree.

Program code : C++

Program GDB link : <https://onlinegdb.com/v8jR1QyGB>

```
#include<iostream>
#include <bits/stdc++.h>
using namespace std;
#define V 5

int findDistance(int distance[],bool check[]){
int minimum= INT_MAX, min_index;

    for (int i = 0; i < V; i++)
        if (check[i] == false && distance[i] < minimum)
            minimum = distance[i], min_index = i;

    return min_index;
}

void printMST(int parent[], int graph[V][V])
{
    cout<<"Edge \tWeight\n";
    for (int i = 1; i < V; i++)
        cout<<parent[i]<<" - "<<i<<" \t"<<graph[i][parent[i]]<<" \n";
}

void printMST(int graph[V][V]){
    // assign distance
    int distance[V];
    //To store the parent visited vertex
    int parent[V];
    //To mark weather vertex is visited or not
    bool check[V];
    for(int i=0;i<V;i++){
        check[i]=false;
        //Assign a very large no. to distance assume infnity
        distance[i]= INT_MAX;
    }
    //as root node of parent won't have any parent
    parent[0]=-1;
    // 0 as root node
    distance[0]=1;
    //To find out the Visited Vertex
    int Edges = V-1;//no of edges in spanning tree
    for(int i=0;i<Edges;i++){
```

```

int minimum = findDistance(distance,check);

check[minimum]= true;

//If we encounter less Value then we haVe to update that
//so
for(int j=0;j<V;j++){
    if(graph[i][j] && check[j]==false && graph[i][j]<distance[j]){
        minimum = graph[i][j];
        parent[j]=i;
    }
}

}

// print the constructed MST
printMST(parent, graph);
}

int main(){
    /* Let us create the following graph

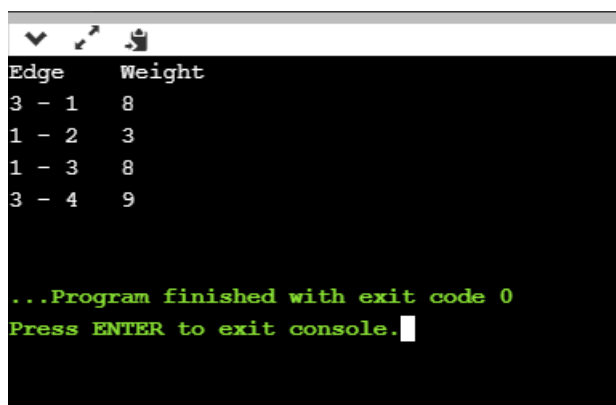
    (0)-2-(1)-3-(2)
    |   |
    6 ---|
    ||-|
    (3)---9---(4)
           */
    int graph[V][V] = { { 0, 2, 0, 6, 0 },
                        { 2, 0, 3, 8, 5 },
                        { 0, 3, 0, 0, 7 },
                        { 6, 8, 0, 0, 9 },
                        { 0, 5, 7, 9, 0 } };

    //To print
    printMST(graph);

return 0;
}

```

Output :



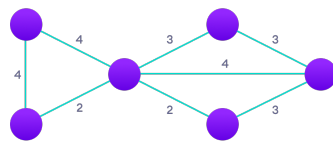
```

Edge    Weight
3 - 1    8
1 - 2    3
1 - 3    8
3 - 4    9

...Program finished with exit code 0
Press ENTER to exit console.

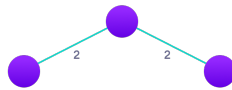
```

In computer science, Prim's (also known as Jarník's) algorithm is a greedy algorithm that finds a minimum spanning tree for a 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.

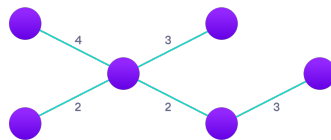


given figure

Step: 1



Step: 4



Step: 6

obtained minimum
spanning tree