

Sardar Patel Institute of Technology

SEM IV: DESIGN AND ANALYSIS OF ALGORITHMS.

Name	SMRUTI SONEKAR
UID no.	2021700064
BRANCH:	SE CSE (DATA SCIENCE)
Experiment No.	6

TOPIC:	PRIM'S ALGORITHM
QUERY:	<p>Prim's algorithm is a minimum spanning tree algorithm that takes a graph as input and finds the subset of the edges of that graph which form a tree that includes every vertex has the minimum sum of weights among all the trees that can be formed from the graph.</p> <p>WORKING OF PRIMS ALGORITHM: (Greedy Algorithm)</p> <ul style="list-style-type: none">• Initialize the minimum spanning tree with a vertex chosen at random.• Find all the edges that connect the tree to new vertices, find the minimum and add it to the tree.• Keep repeating step 2 until we get a minimum spanning tree. <p>PRIM'S algorithm time complexity : $O(E * \log V)$.</p> <p>Prim's Algorithm Application:</p> <ul style="list-style-type: none">• Laying cables of electrical wiring.• In network designed.

- To make protocols in network cycles.

PROGRAM:

```
#include <stdio.h>
#include<stdbool.h>
#include<string.h>
#define INF 999999
#define v 5

int minimumKey(int key[],bool mSet[])
{
    int min=INF;
    int minindex;
    int c;
    for(c=0;c<v;c++)
    {
        if(mSet[c]==false && key[c]<min)
        {
            min=key[c];
            minindex=c;
        }
    }
    return minindex;
}

int printMST(int parent[],int graph[v][v])
{
    printf("Edge \tWeight\n");
    for (int i = 1; i < v; i++)
    {
        printf("%d - %d \t%d \n", parent[i], i,
            graph[i][parent[i]]);
    }

    int sum=0;
    for(int j=1;j<v;j++)
    {
        sum=graph[j][parent[j]]+sum;
    }

    printf("\n Minimum Weight :: %d ",sum);
}
```

```

void prim(int graph[v][v])
{
    int parent[v];
    int k[v];
    bool mSet[v];
    int i,count,p,j;
    for(i=0;i<v;i++)
    {
        k[i]=INF;
        mSet[i]=false;
    }
    k[0]=0;
    parent[0]=-1;

    for(count=0;count<v-1;count++)
    {
        p=minimumKey(k,mSet);
        mSet[p]=true;
        for(j=0;j<v;j++)
        {
            if(graph[p][j] && mSet[j]==false && graph[p][j]<k[j])

            {
                parent[j]=p;
                k[j]=graph[p][j];
            }
        }
    }
    printMST(parent, graph);
}

int main()
{
    int i,j;
    int numedge;

    //printf("No of vertices:");
    //scanf("%d",&v);
    int k[v];
    int G[v][v];
    for(i=0;i<v;i++)

```

```
{  
    for(j=0;j<v;j++)  
    {  
        printf("Enter G[%d][%d]:",i+1,j+1);  
        scanf("%d",&G[i][j]);  
    }  
}  
  
prim(G);  
  
return 0;  
}
```

RESULT:

```
PS D:\c_programming\mudir\daa> gcc prim1.c
PS D:\c_programming\mudir\daa> .\a.exe
Enter G[1][1]:0
Enter G[1][2]:0
Enter G[1][3]:3
Enter G[1][4]:0
Enter G[1][5]:0
Enter G[2][1]:0
Enter G[2][2]:0
Enter G[2][3]:10
Enter G[2][4]:4
Enter G[2][5]:0
Enter G[3][1]:3
Enter G[3][2]:10
Enter G[3][3]:0
Enter G[3][4]:2
Enter G[3][5]:6
Enter G[4][1]:0
Enter G[4][2]:4
Enter G[4][3]:2
Enter G[4][4]:0
Enter G[4][5]:1
Enter G[5][1]:0
Enter G[5][2]:0
Enter G[5][3]:6
Enter G[5][4]:1
Enter G[5][5]:0
Edge    Weight
3 - 1    4
0 - 2    3
2 - 3    2
3 - 4    1

Minimum Weight :: 10
```

CONCLUSION:

Prim's algorithm is one way to find minimum spanning tree and it is applied by c program.