Lab 7 Prims Algorithm

Sushant Bansal
1410110454

Code:

```
#include <stdio.h>
#include <limits.h>
#include <stdlib.h>
#include <time.h>

#define V 5
#define MAX_WEIGHT 10

int minimumValue(int key[], int isVisited[], int noOfNodes)
{
   int min = INT_MAX, min_index;

   for (int v = 0; v < noOfNodes; v++){
     if (isVisited[v] == 0 && key[v] < min){
         min = key[v];
         min_index = v;
     }
   }
   return min_index;
}

int printMSTSum(int parent[], int n, int graph[V][V])
{
       int sum = 0;
       for (int i = 0; i < n; i++){
              sum += graph[i][parent[i]];
       }
    printf("The sum of minimum spanning tree is: %d \n",sum);
}


void primMST(int graph[V][V],int noOfNodes)
{
     int parent[noOfNodes];
     int key[noOfNodes];
     int isVisited[noOfNodes];

     for (int i = 0; i < noOfNodes; i++)
        key[i] = INT_MAX, isVisited[i] = 0;

     key[0] = 0; parent[0] = -1;

    for (int count = 0; count < noOfNodes; count++)
    {
       int u = minimumValue(key, isVisited,noOfNodes);

         for (int v = 0; v < noOfNodes; v++){
              if (graph[u][v] > 0 && isVisited[v] == 0 && graph[u][v] < key[v]){
                     parent[v]  = u;
                     key[v] = graph[u][v];
              }
         }
```

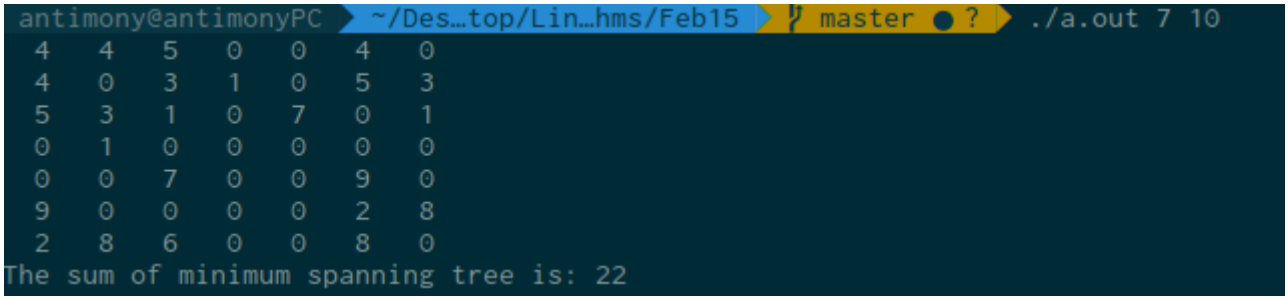
```
      isVisited[u] = 1;
    }
      printMSTSum(parent, noOfNodes, graph);
}

void printG(int arr[V][V],int n){
      for(int i=0;i<n;i++){
            for(int j = 0;j<n;j++){
                  printf("%3d ",arr[i][j]);
            }
      printf("\n");
      }
}

void cleanG(int arr[V][V],int n){
      for(int i=0;i<n;i++){
            for(int j = 0;j<n;j++){
                  if(arr[i][j]<1 || arr[i][j]> MAX_WEIGHT){
                        arr[i][j] = 0;
                  }
            }

      }
}


int main(int argv, char *argc[])
{
      srand(time(NULL));
      int graph[V][V];
      int noOfNodes = atoi(argc[1]);
      int noOfEdges = atoi(argc[2]);
      int i,j,w,k=0;
      while(k<noOfEdges){
            i = rand()%noOfNodes;
            j = rand()%noOfNodes;
            if(graph[i][j] < 1 || graph[i][j] > MAX_WEIGHT){
                  w = (rand()%(MAX_WEIGHT-1)) + 1;
                  graph[i][j] = w;
                  graph[j][i] = w;
                  k++;
            }
            else
                  continue;
      }
      cleanG(graph,noOfNodes);
      printG(graph,noOfNodes);
      primMST(graph,noOfNodes);

    return 0;
}
```

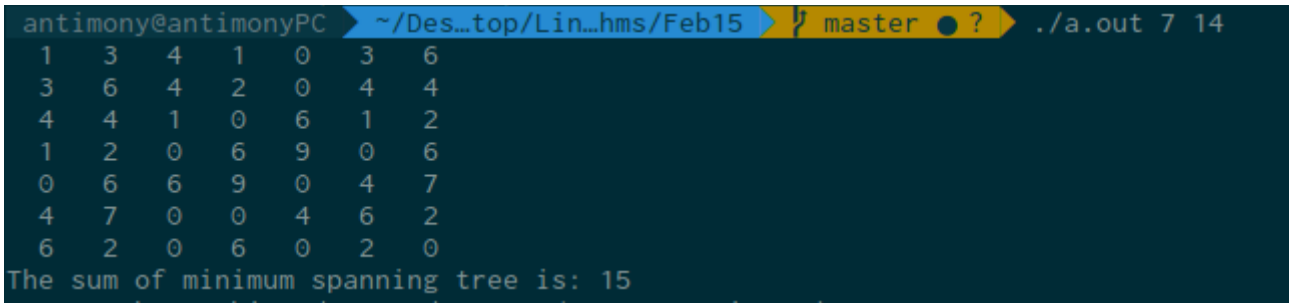
Observations:

Number of Nodes	No of Edges	Cost of MST
5	5	12
	9	14
7	10	22
	14	15

Screenshots:


antimony@antimonyPC: ~/Desktop/Link to Algorithms/Feb15
File Edit View Search Terminal Help
antimony@antimonyPC ~/Des…top/Lin…hms/Feb15 master ● ? ./a.out 5 9
0
6
7
1
3
4
3
3
5
0 0 4 0 0
0 0 5 0 0
4 5 1 1 0
0 0 0 0 0
0 0 0 7 0
The sum of minimum spanning tree is: 14
antimony@antimonyPC ~/Des…top/Lin…hms/Feb15 master ● ?


antimony@antimonyPC ~/Des…top/Lin…hms/Feb15 master ● ? ./a.out 7 10
4 4 5 0 0 4 0
4 0 3 1 0 5 3
5 3 1 0 7 0 1
0 1 0 0 0 0 0
0 0 7 0 0 9 0
9 0 0 0 0 2 8
2 8 6 0 0 8 0
The sum of minimum spanning tree is: 22


antimony@antimonyPC ~/Des…top/Lin…hms/Feb15 master ● ? ./a.out 7 14
1 3 4 1 0 3 6
3 6 4 2 0 4 4
4 4 1 0 6 1 2
1 2 0 6 9 0 6
0 6 6 9 0 4 7
4 7 0 0 4 6 2
6 2 0 6 0 2 0
The sum of minimum spanning tree is: 15