RAJSHAHI UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Lab report: 06
Course No.: CSE 2202

Date of Experiment: 12.12.2018
Date of Submission: 14.01.2019

Submitted to:
Biprodip Pal
Assistant Professor,
Department of Computer Science and Engineering
Rajshahi University of Engineering and Technology

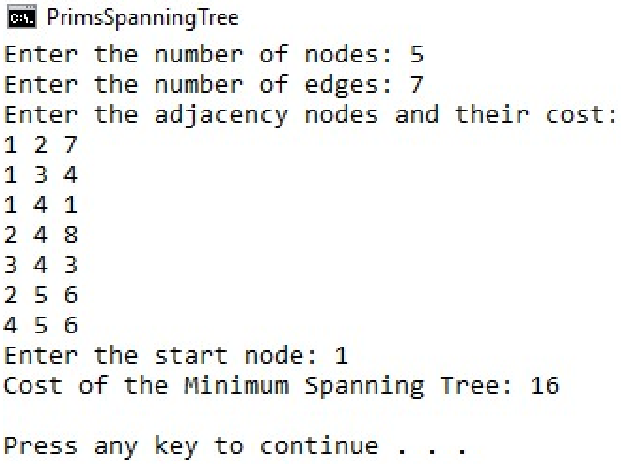
Submitted by:
Riyad Morshed Shoeb
Roll No: 1603013
Section: A
Department of Computer Science and Engineering
Rajshahi University of Engineering and Technology

Problem: Implement Prim's Algorithm to find the minimum spanning tree of a weighed undirected graph.

Code:

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <vector>
#include <queue>
#include <utility>
#define INF 99999
using namespace std;


//class definition
class NODE
{
public:
    int vertex;
    int cost;
    NODE(int, int);
};

NODE::NODE(int _vertex, int _cost)
{
    vertex = _vertex;
    cost = _cost;
}


//global variables
vector < pair <int, int> > V;
priority_queue <NODE> PriorityQueue;
int cost[100];
bool SELECTED[100];


//operator overloading
bool operator>(NODE A, NODE B)
{
    return A.cost > B.cost;
}


//function definition
int PrimsSpanningTree(int StartNode, int NumberOfNodes)
{
    int iterator;
    int SpanningTreeCost = 0;

    for(iterator=0; iterator<NumberOfNodes; iterator++)
    {
        cost[iterator] = INF;
        SELECTED[iterator] = false;
    }
    cost[StartNode] = 0;
    PriorityQueue.push(NODE(StartNode, 0));

    while(!PriorityQueue.empty())
    {
        NODE x = PriorityQueue.top();
        PriorityQueue.pop();
        if(!SELECTED[x.vertex])
        {
            SELECTED[x.vertex] = true;
            SpanningTreeCost += x.cost;
            for(pair<int int> v : V[x.vertex])
            {
                if(!SELECTED[v.first] && (v.second < cost[v.first]))
                {
                    PriorityQueue.push(NODE(v.first, v.second));
                    cost[v.first] = v.second;
                }
            }
        }
    }

    return SpanningTreeCost;
}


//main function
int main(void)
{
    int NumberOfNodes;
    int NumberOfEdges;
    int u;
    int v;
    int cost;
    int iterator;
    int StartNode;

    cout << "Enter the number of nodes: ";
    cin >> NumberOfNodes;
    cout << "Enter the number of edges: ";
    cin >> NumberOfEdges;
    cout << "Enter the adjacency nodes and their cost:" << endl;
    for(iterator=0; iterator<NumberOfEdges; iterator++)
    {
        cin >> u >> v >> cost;
        V[u].push_back(make_pair(v, cost));
        V[v].push_back(make_pair(u, cost));
    }

    cout << "Enter the start node: ";
    cin >> StartNode;

    cout << "Cost of the Minimum Spanning Tree: " <<
PrimsSpanningTree(StartNode, NumberOfNodes) << endl;
}
```

Output:



```
Enter the number of nodes: 5
Enter the number of edges: 7
Enter the adjacency nodes and their cost:
1 2 7
1 3 4
1 4 1
2 4 8
3 4 3
2 5 6
4 5 6
Enter the start node: 1
Cost of the Minimum Spanning Tree: 16

Press any key to continue . . .
```