**DATA ATRUCTURES AND ALGORITHMS- CASE STUDY**

**TEAM- NO NAME IS THE TEAM NAME**

|  |  |
| --- | --- |
| **NAME** | **ROLL NO** |
| Aditi Balaji | CB.EN.U4CSE20303 |
| Kausalyaa Sri | CB.EN.U4CSE20326 |
| Kishore Kumar.V.S | CB.EN.U4CSE20328 |
| Lakshana | CB.EN.U4CSE20332 |

**ABSTRACT**

The aim of this study is to undertake the construction of laying telephone lines between houses and find methods in which it becomes cost effective. Installing cable can be costly especially when there is no mechanism to minimize wastages. This study seeks to establish whether Prim’s can be a reliable method to reduce the cost of installation and maximize profit. The methodology of this study includes using a minimum spanning tree using Prim’s Algorithm. Prim’s method focuses on selecting minimum length between two houses progressively until an optimum route is established. The finding showed that using Prim’s algorithm the size of the cable used can be managed adequately to reduce cost and maximize profit. This study concluded that using the Prim’s method can reduce the overall cost of installing cable since it assists in establishing the minimum route

INTRODUCTION

The minimum spanning tree (MST) will deal with issues in the application fields like that of transportation and telecommunication networks to reduce the costs of distribution in the organization through the best route as the costs become high due to finding spaces, drilling the ground and putting in extension cables. Minimum spanning trees is utilized for networking issues such as cable networks and telephone. These are used to establish the approximate solutions for the complex mathematical issue.

METHODOLOGY

This study used Prim’s algorithm to reduce the cost of the cable for the Internet. The case study is also flexible in such a way that it does not direct the study to use any specific way but allow the study to use different ways to answer “what”, “why” and “how” research questions.

The neighborhood cable connection cannot be analyzed by statistical techniques, instead it needs proper examination of “why” cables need to be installed in particular why and also “what” length of the cable was appropriate. Finally, the study examined “how” the cable was to be installed to ensure there is cost minimization and profit maximization.

CASE ANALYSIS

Installing cable between buildings can be expensive sometimes especially when there is no defined way to follow. The most suitable solution to solving minimum spanning tree is either by using Kruskal or prim algorithm. However, both methods have challenges based on the size of data to be analyzed. The key challenge of using either of the methods includes simplicity and possibilities to avoiding errors. This study will focus on Prim’s algorithm. The methodology of this study is addressed by a case study to show how the Prim’s algorithm can be used in reducing the cable length when the cable connection is established.

To carry out the analysis of the Prim’s algorithm all the possible route and connection to the houses must be identified. After identifying the route, the distance for each route is measured. Once the value of each route is known the analysis is carried out through establishing the minimum vertices progressively until all the houses are connected. If the minimum value appears in multiple times, you can select either of them as long as they do not form a cycle. The process repeated for multiple times until minimum spanning tree is formed.

IMPLEMENTATION

#include<iostream>

using namespace std;

// Number of vertices in the graph

const int V=6;

// Function to find the vertex with minimum key value

int min\_Key(int key[], bool visited[])

{

int min = 999, min\_index; // 999 represents an Infinite value

for (int v = 0; v < V; v++) {

if (visited[v] == false && key[v] < min) {

// vertex should not be visited

min = key[v];

min\_index = v;

}

}

return min\_index;

}

// Function to print the final MST stored in parent[]

int print\_MST(int parent[], int cost[V][V])

{

int minCost=0;

cout<<"Edge \tWeight\n";

for (int i = 1; i< V; i++) {

cout<<parent[i]<<" - "<<i<<" \t"<<cost[i][parent[i]]<<" \n";

minCost+=cost[i][parent[i]];

}

cout<<"Total cost is: "<<minCost;

}

// Function to find the MST using adjacency cost matrix representation

void find\_MST(int cost[V][V])

{

int parent[V], key[V];

bool visited[V];

// Initialize all the arrays

for (int i = 0; i< V; i++) {

key[i] = 999; // 99 represents an Infinite value

visited[i] = false;

parent[i]=-1;

}

key[0] = 0; // Include first vertex in MST by setting its key vaue to 0.

parent[0] = -1; // First node is always root of MST

// The MST will have maximum V-1 vertices

for (int x = 0; x < V - 1; x++)

{

// Finding the minimum key vertex from the

//set of vertices not yet included in MST

int u = min\_Key(key, visited);

visited[u] = true; // Add the minimum key vertex to the MST

// Update key and parent arrays

for (int v = 0; v < V; v++)

{

// cost[u][v] is non zero only for adjacent vertices of u

// visited[v] is false for vertices not yet included in MST

// key[] gets updated only if cost[u][v] is smaller than key[v]

if (cost[u][v]!=0 && visited[v] == false && cost[u][v] < key[v])

{

parent[v] = u;

key[v] = cost[u][v];

}

}

}

// print the final MST

print\_MST(parent, cost);

}

// main function

int main()

{

int cost[V][V];

cout<<"Enter the number of houses:";

for (int i=0;i<V;i++)

{

for(int j=0;j<V;j++)

{

cin>>cost[i][j];

}

}

find\_MST(cost);

return 0;

}

COMPLEXITY ANALYSIS

The time complexity is O(VlogV + ElogV) = O(ElogV), since |E|>=V. It can be improved using Fibonacci Heaps to O(E + V logV).

CONCLUSION

This study demonstrates that the Prim’s algorithm optimal solution is conducted by a procedural selecting minimum length in ascending order. The main rule of the process is to avoid forming a cycle. With Prim’s algorithm, it was clearly shown that the process reduces the cost, especially when compared to a situation where the cable is installed without any proper procedures.