NEW HORIZON COLLEGE OF ENGINEERING

New Horizon Knowledge Park, Ring Road, Marathalli

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC

Accredited by NAAC with 'A' Grade, Accredited by NBA

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

A

MINI PROJECT REPORT

ON

"PROFIT ORIENTED TRANSPORTATION"

Submitted in the partial fulfillment of the requirements in the 5th semester of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

BY

VARNA MURALI

1NH17IS126

Under the guidance of

Mrs. KAVITHA K K,

Sr. Assistant professor,

Dept. of ISE, NHCE

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING

(Autonomous College Permanently Affiliated to VTU, Approved by AICTE, Accredited by NBA & NAAC with 'A' Grade)

New Horizon Knowledge Park, Ring Road, Bellandur Post, Near Marathalli,

Bangalore-560103, INDIA

NEW HORIZON
COLLEGE OF ENGINEERING
New Horizon Knowledge Park, Ring Road, Marathalli
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

I hereby certify that, the report entitled **"Profit Oriented Transportation"** as a part of Mini Project Component in partial fulfillment of the requirements during 5th semester Bachelor of Engineering in Information Science and Engineering during the year 2019-2020(Aug 2019-Nov 2020) is an authentic record of my own work carried out by Varna Murali (1NH17IS126), a bonafied student of NEW HORIZON COLLEGE OF ENGINEERING.

Name & Signature of Student

(Ms. Varna Murali)

Name & Signature of Guide

(Mrs. Kavitha K K)

Name & signature Of HOD

(Dr. R J Anandhi)

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this project. I would like to take an opportunity to thank them all.

First and foremost I thank the management, **Dr. Mohan Manghnani,** Chairman, New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

I would like to thank **Dr. Manjunatha**, Principal, New Horizon College of Engineering, Bengaluru, for his constant encouragement and facilities extended to us towards completing my project work.

I extend my sincere gratitude to **Dr. R J Anandhi**, Head of the Department, Information Science and Engineering, New Horizon College of Engineering, Bengaluru for her valuable suggestions and expert advice.

I deeply express my sincere gratitude to my guide **Mrs. Kavitha K K, Sr. Assistant Professor**, Department of ISE, NHCE, Bengaluru, for her able guidance, regular source of encouragement and assistance throughout this project.

I thank my Parents, and all Faculty members of Department of Information Science and Engineering for their constant support and encouragement.

Last, but not the least, I would like to thank my peers and friends who provided me with valuable suggestions to improve my project.

Varna Murali

(1NH17IS126)

ABSTRACT

Meeting the needs of rural people and thinking through what could potentially affect those who work and live in the rural areas is vital for local authorities. The minimum spanning tree problem has important applications in network design which has been extensively studied in literature. The minimum spanning tree problem on a graph with edge costs and vertex profits asks for a sub tree maximizing the difference between the total cost of all edges in the sub tree and the total profits of all vertices contained in the sub tree.

Minimum spanning tree problem appears in the design of utility networks (e.g. bus services, electrifications) where villages and the network connecting them have to be chosen in the most profitable way. This project demonstrates the application of Prim's algorithms to the design of local access networks. Our main contribution is an efficient algorithmic implementation using an undirected graph model. A case study is analyzed where interesting results are obtained. The minimum spanning tree problem on a graph with edge costs and vertex profits asks for a sub tree maximizing the difference between the total cost of all edges in the sub tree and the total profits of all vertices contained in the sub tree. . By finding the minimum path the social amenities like electricity tap water can be provided more efficiently. The least distance between villages is calculated using prims algorithm. From the rate of fuel the total travel cost is calculated. The total minimum distance is calculated by summing the distances acquired from the minimum distance graph which is determined by Prims algorithm.

By using this project we can increase the transportation facilities in rural areas. It gives information which serves as a guide for provision of social amenities like tap water, electricity, medical services (immunization programs) in terms of the shortest route to take in order to make the services cost effective. It helps to improve the transportation facilities in rural areas which in turn help in the development of the country. Minimum spanning tree is used because they can be computed quickly and easily, and they create a sparse sub graph that reflects a lot about the original graph.

TABLE OF CONTENTS

Acknowledgement	**i**
Abstract	**ii**
Table of Contents	**iii**
List of Figures	**viii**
List of Tables	**ix**
Chapter 1	
Introduction	**1**
1.1 Purpose of Study	2
1.2 Problem Statement	2
1.3 Motivation	3
1.4 Methodology	3
Chapter 2	
System Requirements and language used	**4**
2.1 Hardware and Software Requirements	4
2.2 About Language	5
Chapter 3	**6**
3.1 Architecture	6
3.2 Algorithm	7
3.3 Flowchart	8
3.4 Code	9
Chapter 4	**20**
4.1 Summary	20
4.2 Output Screenshots	21
Chapter 5	
5.1 Conclusion	27
References	**28**

LIST OF FIGURES

Figure No.	Figure Name	Page No.
3.1	Architecture of profit oriented transportation	6
3.2	Flowchart of profit oriented transportation	8
4.1	Asking the user to enter total set of villages	21
4.1	Asking the user to enter the pair of villages	22
4.2	Displaying the given data	23
4.3	calculating the minimum distance using spanning tree	24
4.4	Calculating the cost of travelling	25
4.5	Asking the user if he wants to continue	

Chapter 1

INTRODUCTION

Profit oriented transport system is a project in which the travelling needs of the people are met. It is used for finding the shortest path take for visiting two different cities. The shortest path and time taken to travel that distance can be determined from this project. It saves the time and reduces expenses. The method used to find the shortest distance is minimum spanning tree using prims algorithm. The minimum spanning tree problem has important applications in network design. The minimum spanning tree problem on a graph with edge costs andvertexprofits asks forasubtree maximizing the difference between the total cost of all edges in the sub tree and the total profits of all vertices contained in the sub tree. Minimum spanning tree problem appears in the design of utility networks (e.g. bus services, electrifications) where villages and the network connecting them have to be chosen in the most profitable way. They can be computed quickly and easily, and they create a sparse sub graph that reflects a lot about the original graph. They provide a way to identify clusters in sets of points.

With the help of this project we can increase the transportation facilities of rural areas. By finding the optimal path we can provide facilities to villages in a cost effective way. This algorithm is very effective in determining the shortest distance between the set of villages. It reduces the cost offuel and time for transportation of passengers from one place to another.

Recent advances in technology have made network design for connecting villages economically feasible. The method can also be explored by other organizations willing to achieve their goals with a reduced cost and effective use of their resources especially in areas of electrical installation, pipe water distribution and telecommunication.

The Minimum Spanning Tree (MST) problem is another model of a spanning tree. A spanning tree T of G is a connected acyclic subgraph that spans all the nodes. Every spanning tree of G has n-1 arcs (where n is a number of nodes), which is a characteristic property of a tree. Also, a tree has at least two leaf nodes (i.e. nodes with degree 1), and every two nodes of a tree are connected by a unique path.

1.1 Purpose of Study

This project determines the shortest path to reach a particular destination and even determines the cost of travel. In the present system traveling expenses required for reaching a particular place cannot be determined beforehand and we may not be able to choose the optimum path always. This back draw can be overcome using this project. Recent advances in technology have made network design for connecting villages economically feasible. In a typical planning scenario, the input is a set of villages together with the distances between them. Cost of network is that of fuel used in traveling round the villages. With minimum spanning tree we can find the optimum distance between villages and save the cost of fuel and the time required for travelling. It provides information for social amenities like electricity, tap water in terms of the shortest route to take to make the service cost effective. The motivation of this project is to provide improved network within the rural networks. The optimum path can be determined between a set of villages with the help of this project. This saves time and cost of travel.

1.2 Problem Statement

- A minimum cost path should be determined
- There is a need to start from a particular village and end in a particular village

1.3 Motivation of Project

This project determines the shortest path to reach a particular destination and even determines the cost of travel. The motivation of this project is to provide improved network within the rural networks. The optimum path can be determined between a set of villages with the help of this project. This saves time and cost of travel. In the current system the travel expenses cannot be determined beforehand. Good transport is a key to improving access to services. With the help of this project we can find the optimum path which will help in providing social amenities to villages.

1.4 Methodology

To implement a profit oriented transportation system using Prims algorithm. The set of villages with their distances are given as input. The minimum edge is added to the spanning tree and a graph is generated connecting the villages. The total distance to be covered is determined. The cost of the fuel is taken as input. Then the cost required for travelling the total distance is determined. From this project the total expenses of the travel can be determined beforehand and the transport facilities in rural areas can be improved. The language used to implement this project is c++. And the algorithm used is Prim's algorithm which 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. It can be computed quickly and efficiently. The input is a set of villages together with the distances between them. Cost of network is that of fuel used in traveling round the villages.

Chapter 2

SYSTEM REQUIREMENT SPECIFICATION

A System Requirements Specification (SRS) (also known as a Software Requirements Specification) is a document or set of documentation that describes the features and behavior of a system or software application. Usually a combination of problems and opportunities are needed to provide motivation for a new system.

2.1 Hardware and Software Requirements

Processor: Intel Core i3-380M dual-core processor

Ram: 1GB (32 bit) o2 3GB (64 bit)

Hard disk: 16GB

Software System Configuration:

Operating System: windows

Programming Language: C++

Compiler: Code blocks

2.2 About the Language

C++ is a general-purpose programming language created by Bjarne Strostrup as an extension of the C programming language, or "C with Classes". The language has expanded significantly over time, and modern C++ has object-oriented, generic, and functional features in addition to facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers, including the Free Software Foundation, LLVM, Microsoft, Intel, Oracle, and IBM, so it is available on many platforms. C++ was designed with a bias toward system programming and embedded, resource-constrained software and large systems, with performance, efficiency and flexibility of use as its design highlights. C++ was designed with a bias toward system programming and embedded, resource-constrained software and large systems, with performance, efficiency, and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce, Web search, or SQL servers), and performance-critical applications (e.g. telephone switches or space probes).

Chapter 3

SYSTEM DESIGN

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. This chapter consists of the architecture, flowchart and algorithm of the project.

3.1 Architecture

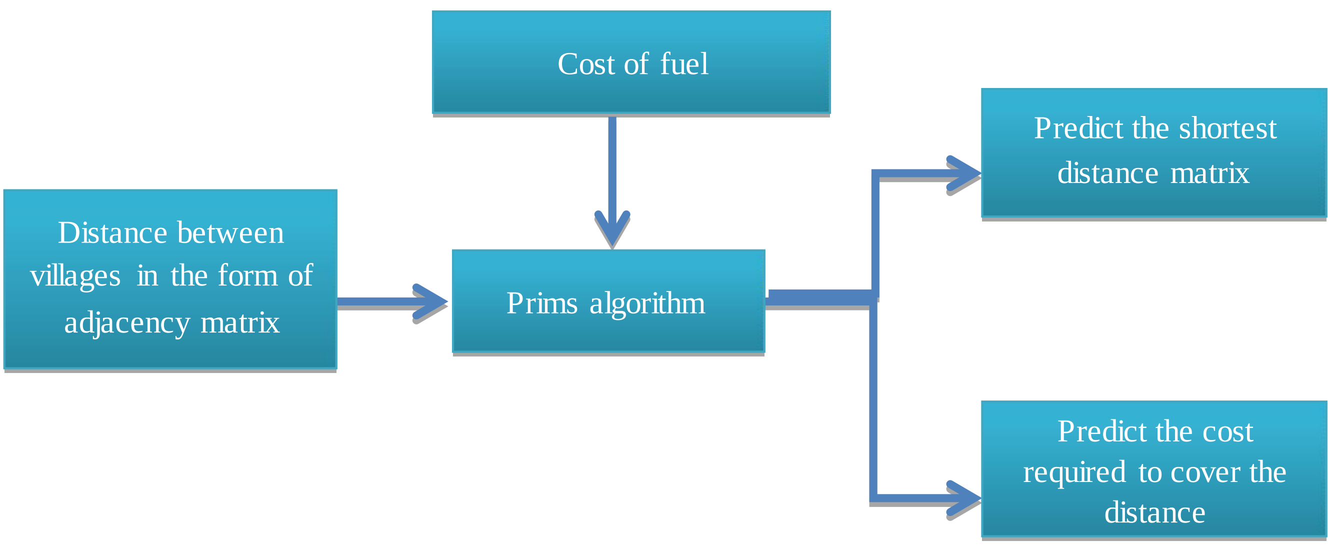

Figure 3.1 Architecture of Profit Oriented Transport System

The basic architecture of the profit oriented transport system is shown in the above figure. Profit oriented transportation is a project which uses prim's algorithm to find the minimum path and then calculate the total cost required to travel that distance. Using this project the transportation facilities of the rural places can be improved. The basic facilities like electricity, tap water can be in terms of the shortest route to take in order to make the services cost effective.

3.2 Algorithm

Step 1: start

Step 2: input the total number of villages

Step 3: Input the pairs of villages and distance between them

Step 4: Add the minimum node of the given graph to minimum spanning tree

Step 5: Calculate the minimum distance pairs of villages using prims algorithm of minimum

Spanning tree

Step 6: input the cost of fuel for 1 km distance

Step 7: display the total distance between villages.

Step8: display the costrequired totravel the total distance displayed in step 7.

Step 9: print 1 to insert

If true goto step 3

Step 10: stop

3.3 Flowchart

Figure 3.2 depicts the diagrammatic representation of the algorithm. First the total number of villages is taken as input from the user. Then the user is asked to enter the villages in pairs with the distance between the villages. The least distance is calculated using the minimum spanning tree algorithm. The least distance between villages is calculated in such a way that each village is visited only once. The cost of the fuel is taken as input and the total distance and the cost is calculated. The user is asked whether he wants to insert a new pair of village. If yes then the new pair is added to the graph and the new cost is calculated.

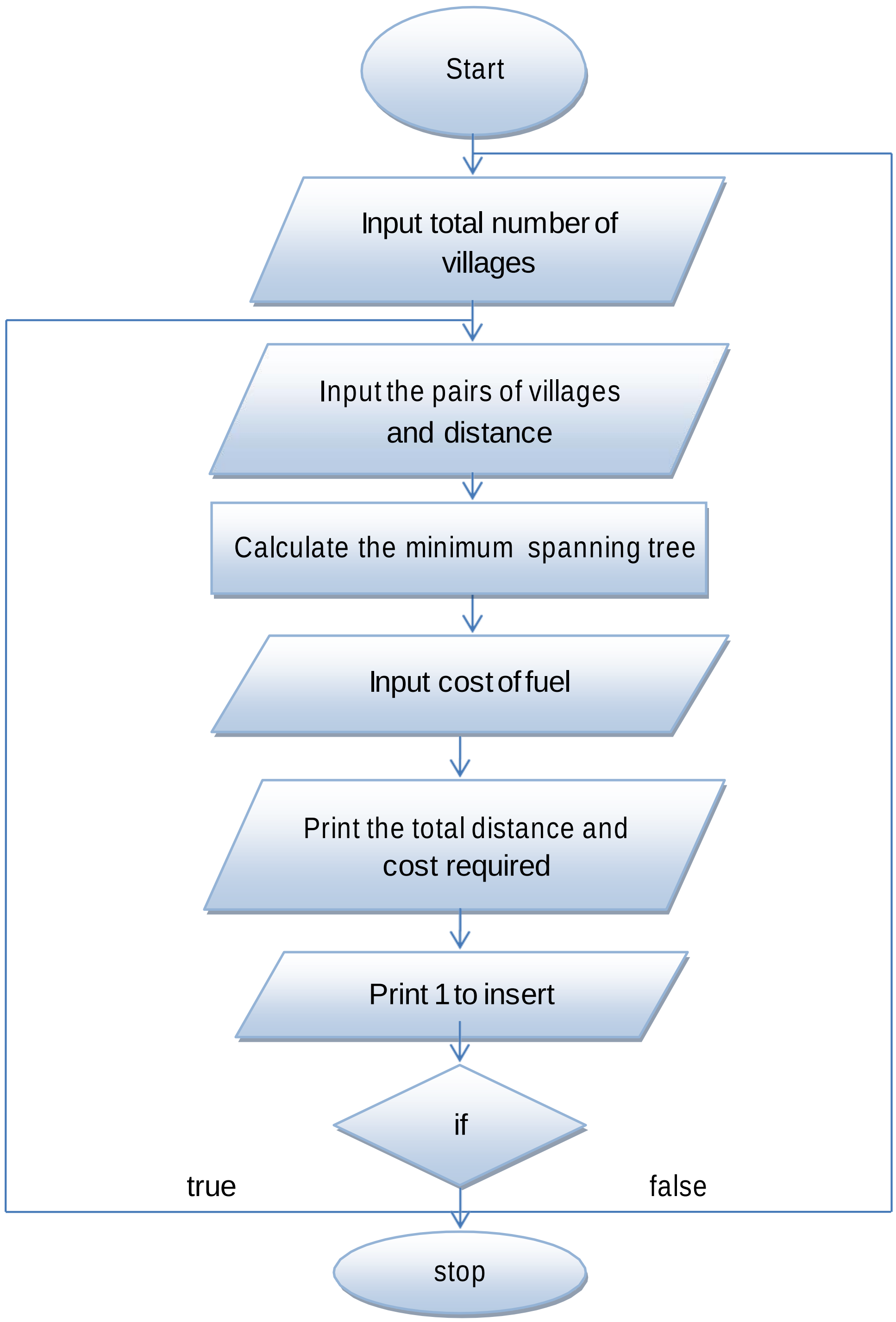

Figure 3.2 Flowchart of profit oriented transportation

3.4 code and implementation

```
#include<iostream>

#include<conio.h>

using namespace std;

#include<string.h>

struct edge

{

    char h1[10],h2[10];

      int weight;

};

struct state

{

      char name[100];

};

struct edge edges[100];

struct edge spanning_tree[100];

int no_of_edges=0,no_of_edges_of_spanning_tree=0,i;

int index_of_edge_to_add;

void get_graph();
```

```
void print_the_system();

int index_of_lowest_marked_edge();

void prim();

void add_lowest_marked_edge_to_spanning_tree(int);

void add_new_state(int);

void initialise();

void calc();

struct state covered_states[200];

int no_of_states=0;

void get_graph()

{

        int n;

        cout<<"\nEnter the total sets of villages in pairs\n";

        cin>>n;

        cout<<"\n enter pair of village that are linked together\n";

        for(no_of_edges=0;no_of_edges<n;no_of_edges++)

        {

            cin>>edges[no_of_edges].h1;

            cin>>edges[no_of_edges].h2;
```

```
        //getchar ();
        cin>>edges[no_of_edges].weight;
          //getchar();
          //no_of_edges++;
    }
}
void print_the_system()
{
    cout<<"\n\t ...................................\n\t";
    cout<<"\n the given data is\n";
    for(i=0;i<no_of_edges;i++)
    cout<<"\n"<<i<<"\t"<<edges[i].h1<<"\t"<<edges[i].h2<<"\t"<<edges[i].weight<<endl;
    cout<<"\n least distances calculated"<<endl;
    for(i=0;i<no_of_edges_of_spanning_tree;i++)
    cout<<"\n"<<spanning_tree[i].h1<<"\t"<<spanning_tree[i].h2<<"\t"<<spanning_tree[i].
weight;
}
void add_lowest_marked_edge_to_spanning_tree(int  index_of_edge_to_add)
{
```

```
    strcpy(spanning_tree[no_of_edges_of_spanning_tree].h1,edges[index_of_edge_to_add
].h1);

    strcpy(spanning_tree[no_of_edges_of_spanning_tree].h2,edges[index_of_edge_to_add
].h2);

spanning_tree[no_of_edges_of_spanning_tree].weight=edges[index_of_edge_to_add].weight;

no_of_edges_of_spanning_tree++;

}

void add_new_state(int index_of_edge_to_add)

{

    int b1=1,b2=1;

    for(i=0;i<no_of_states;i++)

    {

        if(strcmp(covered_states[i].name,edges[index_of_edge_to_add].h1)==0)

            b1=0;

            if(strcmp(covered_states[i].name,edges[index_of_edge_to_add].h2)==0)

            b2=0;

    }

            if(b1==1)

            {

            strcpy(covered_states[no_of_states].name,edges[index_of_edge_to_add].h1);
```

```
                no_of_states++;
                }
                if(b2==1)
                {

        strcpy(covered_states[no_of_states].name,edges[index_of_edge_to_add].h2);
        no_of_states++;
                }
}
int index_of_lowest_ marked_edge()
{
        int j;
        int b1,b2;
        int min;
        int the_lowest_marked_edge;
        int marked_edges_indexes[100],no_of_marked_edges_indexes=0;
        for(i=0;i<no_of_edges;i++)
        {
                b1=1;
                b2=1;
```

```
//int("%d",no_of_states);
for(j=0;j<no_of_states;j++)
{
    if(strcmp(covered_states[j].name,edges[i].h1)==0)
    b1=0;
    if(strcmp(covered_states[j].name,edges[i].h2)==0)
    b2=0;
}
if((b1==0&&b2==1)||(b1==1&&b2==0))
{
    marked_edges_indexes[no_of_marked_edges_indexes]=i;
    no_of_marked_edges_indexes++;
}
}
if(no_of_marked_edges_indexes==0)
{
    return -1;
}
//deleting the lowest marked edge
```

```
        min=edges[marked_edges_indexes[0]].weight;
        the_lowest_marked_edge=marked_edges_indexes[0];
        for(i=0;i<no_of_marked_edges_indexes;i++)
         {
         if(edges[marked_edges_indexes[i]].weight<min)
         {
                min=edges[marked_edges_indexes[i]].weight;
                the_lowest_marked_edge=marked_edges_indexes[i];
         }
   }
    return the_lowest_marked_edge;
}
void prim()
{
         //int index_of_edge_to_add;
         index_of_edge_to_add=index_of_lowest_marked_edge();
         if(index_of_edge_to_add==-1)
         return;
         add_lowest_marked_edge_to_spanning_tree(index_of_edge_to_add);
```

```
        add_new_state(index_of_edge_to_add);
        print_the_system();
  prim();
}
void initialise()
{
   get_graph();
        strcpy(covered_states[no_of_states].name,edges[0].h1);
        no_of_states++;
        prim();
        calc();
}
void calc()
{
   int rate,sum=0,km=0;
   cout<<"\nEnter Rate of the fuel in rupees\n";
   cin>>rate;
   for(i=0;i<no_of_edges_of_spanning_tree;i++)
     km+=spanning_tree[i].weight;
```

```
    cout<<"\n the total minimum  distance between all houses"<<km;

    for(i=0;i<no_of_edges_of_spanning_tree;i++)

      sum+=(spanning_tree[i].weight*rate);

     cout<<"\n cost of traveling would be\n"<<sum;

}

void insert()

{

   no_of_states=0;

   no_of_edges_of_spanning_tree=0;

   cin>>edges[no_of_edges].h1>>edges[no_of_edges].h2>>edges[no_of_edges].weight;

   no_of_edges++;

   strcpy(covered_states[no_of_states].name,edges[0].h1);

          no_of_states++;

          prim();

          calc();

}

int main()

{

    int ch;
```

```
   char c;

cout<<"\n$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$\n";

cout<<"\n DESCRIPTION: This program basically calculates\n the total minimum distance
between villages\n and the cost of fuel when the distance between\n houses that are linked
together is given\n";

        initialise();

        do

  {

     cout<<"press 1 to insert houses in pairs\n";

     cout<<"enter choice\n";

     cin>>ch;

     if(ch==1)

     {

        cout<<"you are asked to insert the pair of houses along with the distance\n";

        insert();

     }

 cout<<"do you want to continue to calculate for another set of houses?\npress y to
continue\n press any other key to quit\n ";

        c=getch();

 }while(c=='y'||c=='Y');
```

```
return 0;

}
```

Chapter 4

Results and Discussion

Results and discussion is an important role in documentation. This chapter the summary of the results obtained and output of the project. From this chapter the idea of the project and what the code actually does can be determined. A clear idea about the code and its output will be received from this chapter.

4.1 Summary of results obtained:

This project uses prims algorithm to calculate the minimum distance between villages. This project is used to increase the transportation facilities in rural areas. By finding the minimum path the social amenities like electricity tap water can be provided more efficiently. The least distance between villages is calculated using prims algorithm. From the rate of fuel the total travel cost is calculated. The total minimum distance is calculated by summing the distances acquired from the minimum distance graph which is determined by Prims algorithm.

4.2 output (snapshots)

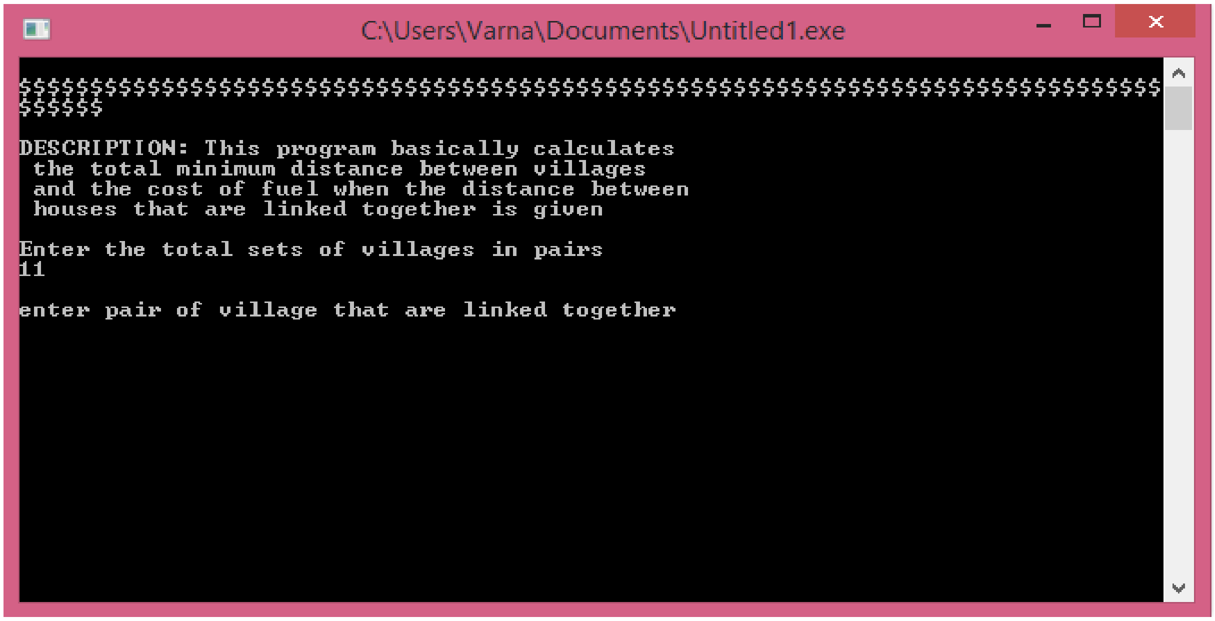

Figure 4.1 asking the user to enter total set of villages

In the above figure the description of the project is given and the user is asked to enter the total set of villages. It is the total number of villages that is linked together for which we have to find the minimum spanning tree.

Figure 4.2 Asking the user to enter the pair of villages

After entering the total set of villages the user is asked to enter the villages connected to each other and the distance between them as shown in figure 4.2. The user will have to enter two villages which are connected to each other and the distance between them. From these set of villages the least distance graph will be determined.

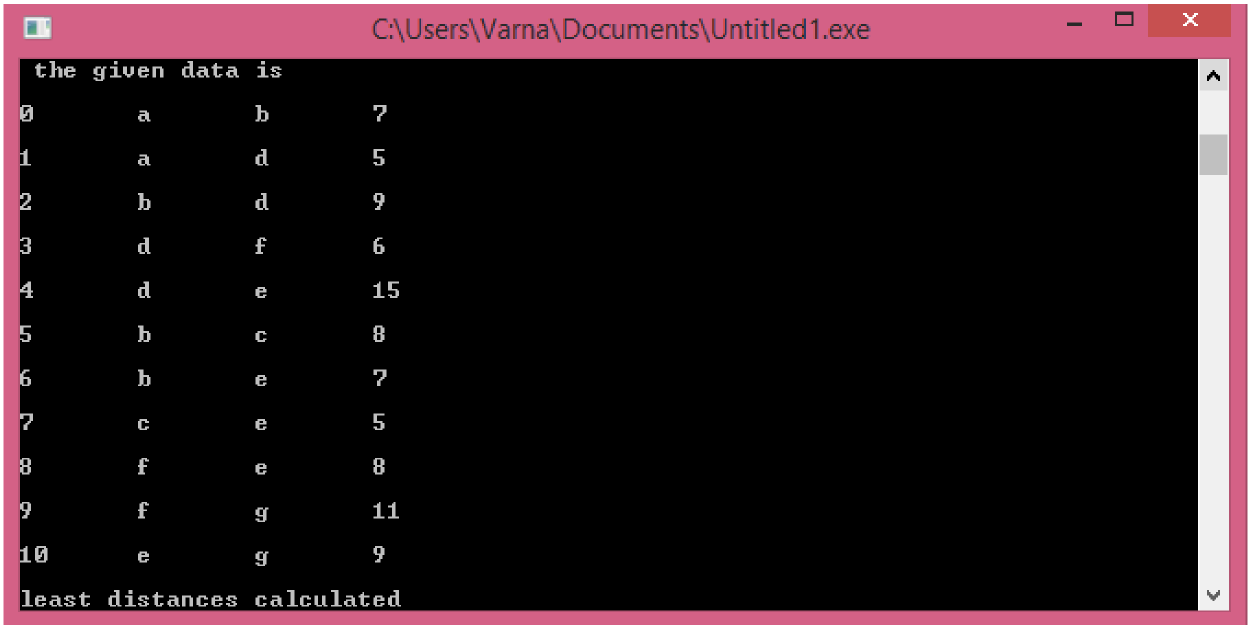

Figure 4.3 Displaying the given data

The data which is given as input by the user is displayed in the above figure. The user will have to enter two village names and the distance between the villages. This data will be displayed in the same format. All the villages taken as input from the user will be displayed in the same order.

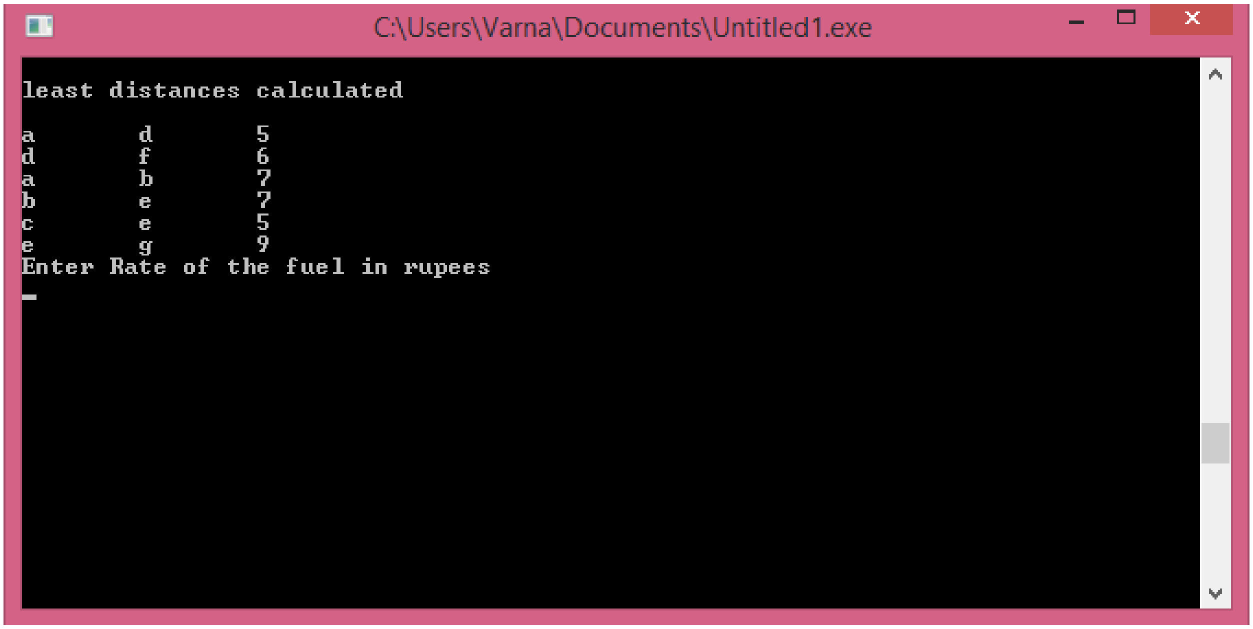

Figure 4.4 Calculating the minimum distance using spanning tree

The minimum distance from all the pairs of given villages is determined using prims algorithm and is added to the spanning tree which is acyclic. Each village will be visited only once. The first village, the second village and the minimum distance between them is printed.

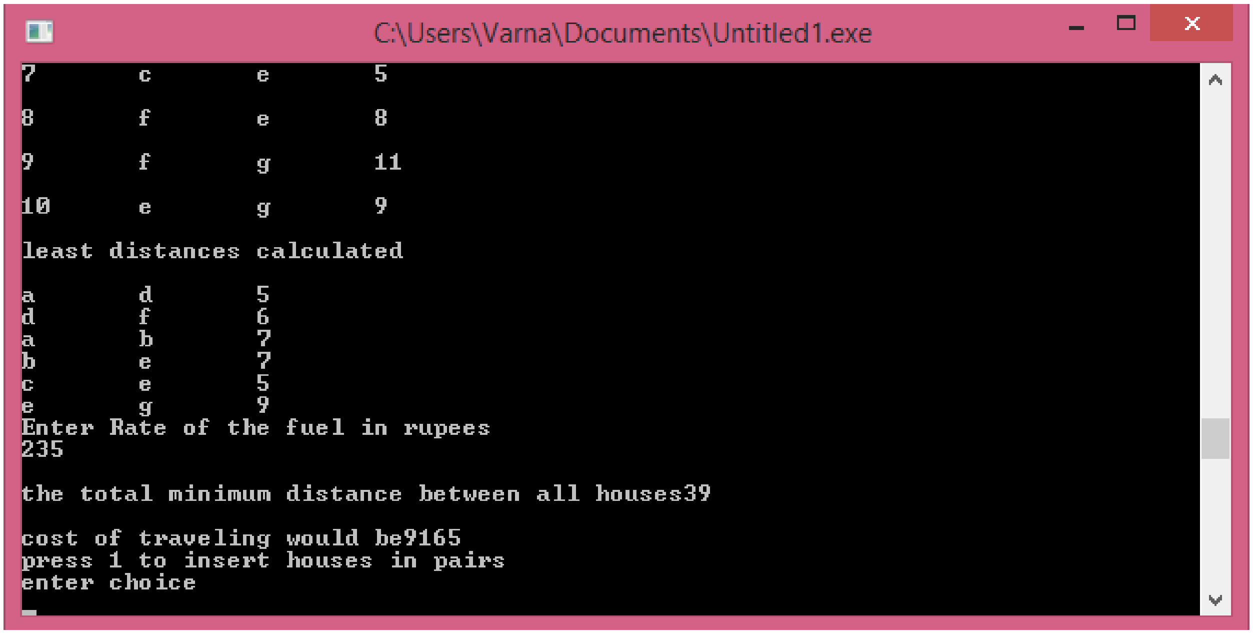

Figure 4.5 calculating the cost of travelling

In the figure 4.5 the user is asked to enter the cost of fuel. From that the total minimum distance that is to be travelled and the total amount required for travelling that distance is calculated. The total distance to travel is calculated by summing up all the distances that is determined by using prims algorithm.

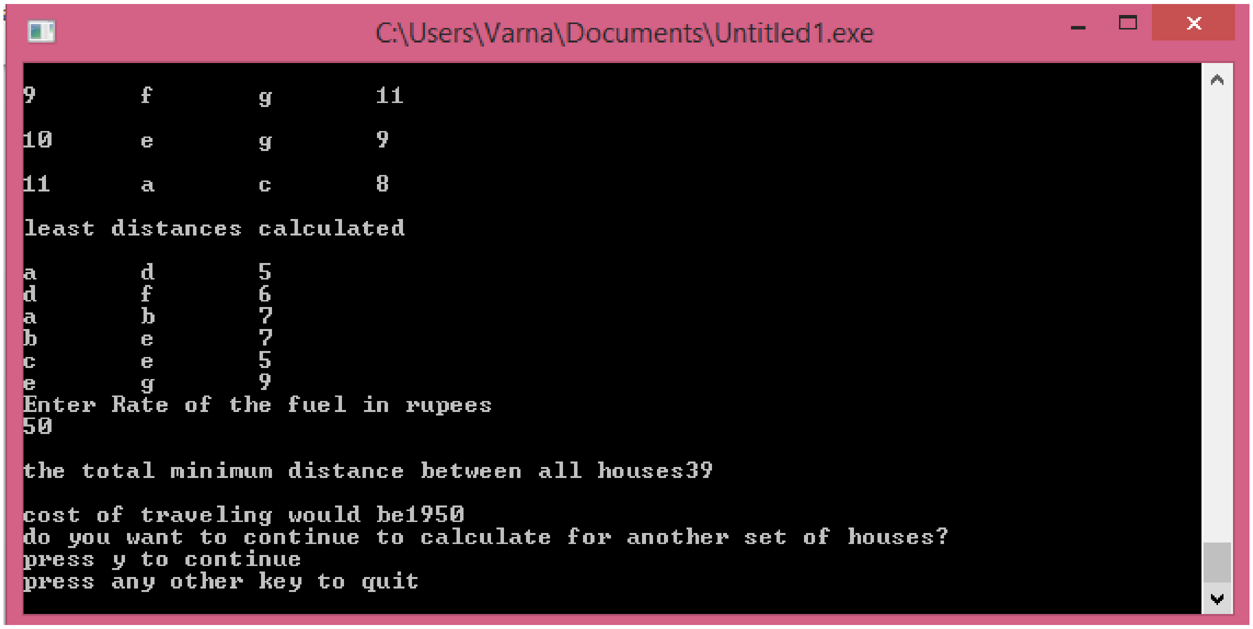

Figure 4.6 Asking the user if he wants to continue

The user is asked if he wants to insert a new pair of village to current pair of villages. If the user wants to enter then the new pair of village is inserted and the new total distance and new cost is determined. After that the user is asked if he wants to continue. If the user wants to then the figure 4.1 is repeated.

Chapter 5

CONCLUSIONS

This algorithm is very effective in determining the shortest distance between the set of villages. It reduces the cost of fuel and time for transportation of passengers from one place to another. With minimum spanning tree we can find the optimum distance between villages and save the cost of fuel and the time required for travelling. It provides information for social amenities like electricity, tap water in terms of the shortest route to take to make the service cost effective. It will help transport service providers on accessibility issues. Good transport is key to improving access to services. It helps to improve the transportation facilities in rural areas which in turn help in the development of the country. Minimum spanning tree is used because they can be computed quickly and easily, and they create a sparse sub graph that reflects a lot about the original graph.

REFERENCES

[1] www.researchgate.net/publication/

[2]www.geeksforgeeks.org

[3]Introduction to design and analysis of algorithms by Anany Levitin