Data Structures and Algorithms

Project Report

Topic: Railway Reservation Mapping

**Preparer:**

Name: Arnav Raviraj

Reg No: 14BEC0853

**Problem Statement:**

This project is about railway reservation mapping which input the various routes with intermediate stations and their fare without the previous station. We can compare the fares of the same route and we calculate the fare for a particular part of the route.

For example, if there are trains going from Bangalore to New Delhi, stopping at Panaji, Pune, Mumbai, Ahmedabad and Jaipur, and a person desire to go from Mumbai to Jaipur then it will display fare only from Mumbai to Jaipur and would display all the train from which he/she can travel.

**Data Structure used:**

We have used array of link list so that we can store different routes as a form of link list in the array.

The most important feature that we have implemented is the use of recursion. Without recursion, our code would have been 5-6 times larger and very tough to implement.

**Functions:**

We have an allocate function which stores the all the stations of a particular route. This function works on recursion.

Then we have a search function which inputs starting and end station of the route that the user wants to search. We have a smart mapping function which checks whether starting and ending station are available are not by calling another function with the help of recursion and calculate the fare.

**List of Errors:**

• One of the error that we faced was that, when we were calculating the fare between two stations the program was calculating the whole fare from the main starting to the ending point which we entered.

We solved this error by subtracting the fare from main starting

point of the route till the starting point which we entered at the time of searching.

• Other error which we faced was that the program was not able to

identify if the stations which we entered during searching were present in our root or not.

We solved this by taking a flag variable which is initially zero and

becomes 1 if the stations are in the root. So, we were able to show

“No routes found” if the stations were not there in the root.

**Reference:**

• GeeksforGeeks.com

• Tutorialspoint.com

**Code:**

/\* DSA Project

Railway reservation (smart mapping)

Name : Arnav Raviraj

Reg no: 14BEC0853

\*/

#include <iostream>

#include <string> using namespace std; int no;

class railway

// create a class railway

{

public:

railway \*next;

string stname;

//name of the station

int ffp;

//fare from prev

void allocate (int \*f)

//allocating a particular route in array of link list(recursive function)

{

if(\*f!=0)

{

cout<<"\n";

next=new railway;

--\*f;

cout<<"\t\tEnter the name of the station : ";

cin>>stname;

cin>>ffp;

cout<<"\t\tEnter cost from prev station : ";

next->allocate(f);

}

else

{

next=NULL;

}

}

int check(string st ) //Checking whether station exist or not in the route

{

if(st==stname)

{

return(1); //found

}

if(next==NULL)

return(-1); //not found return(next->check(st));

}

int faref(string st) //Calculating the fare

{

if(st==stname)

return(ffp);

return(next->faref(st)+ffp); //recursive

}

};

void sm(railway route,string st,string ed,int \*nc) //smart mapping function (check the start point , end point , clculates fare and displays the route

{

if(route.check(st)==1)

{

railway \*r=&route;

while(r->stname!=st)

{

r=r->next;

}

if(r->check(ed)==1)

{

\*nc=1;

cout<<"\n\t\tTotal Fare of the route: "<<(r->faref(ed)-(r-

>ffp))<<endl;

cout<<"\t\tThe route is ";

while(r->stname!=ed)

{

cout<<r->stname<<" ---> ";

r=r->next;

}

cout<<r->stname<<"\n\n";

}

}

}

void searchh(railway route[100]) //Taking input of start and end point and calling smart mapping function

{

string st;

cout<<"\t\tEnter the start point : ";

cin>>st;

cout<<"\t\tEnter end point :";

string ed;

cin>>ed;

int \*nc=new int;

\*nc=0;

for(int i=0;i<no;i++)

sm(route[i],st,ed,nc);//smart mapping

if(\*nc==0)

cout<<"\n\t\tNo routes found \n\n";

}

int main() //main function

{

int flag=0;

railway route[100];

string name\_route[100];

no=0;

while(flag!=1)

{

int n;

cout<<"\n\n\n\t\t\t\t\t\tRailway Reservtion Mapping\n\n";

cout<<"\n\n\t\t1.New Route"; cout<<"\n\t\t2.Search route and get fare"; cout<<"\n\t\t3.Exit";

cin>>n;

switch(n)

{

case 1:

{

no++;

cout<<"\n\n\t\tEnter the number of stations : ";

int \*m=new int;

cin>>(\*m);

cout<<"\t\tEnter the route name : ";

//identity of each route cin>>name\_route[no];

route[no-1].allocate(m);

//allocating routes in array of link list

}

break;

case 2:

searchh(route);

//Calls search function

break;

case 3:

break;

flag=1;

default:

cout<<"\t\tWrong Input\n";

break;

}

}

return 0;

}