PROJECT REPORT ON

DATA STRUCTURES AND ALGORITHMS

TITLE OF THE PROJECT-

Bus Availability Check

System in Punjab

SUBMITTED BY-

Animesh Attri (101603032)

Ankit Madan (101603040)

Anmol Jain (101603045)

Anmol Mahajan (101610012)

Submitted to the

Computer Science & Engineering Department

Thapar Institute of Engineering and Technology, Patiala

May 2018

TABLE OF CONTENTS

PROBLEM FORMULATION

In these days where travelling via buses is so much in demand and there is no such platform for the Punjab roadways where travellers can find the best possible route to reach from one source to destination, we decided to make this possible in our project to help all the travellers.

The problem in Punjab is that there is no raw data available to the travellers about the buses i.e., when the buses start and when they will reach approximately if no misshapen occurs. So we collected the data from google which includes the starting time of buses, their reaching time and the number of buses from one city to another.

After this we decided to ask the user at what time he she wants to leave that particular city. So our program tells the user at what time he/she should leave the starting city based on the input time entered by him/her. The code tells the user the best possible route and approximately the time by which he/she would reach the destination.

Next problem faced in Punjab is the cost of travelling..!! There is no such software or database that tells the traveller about the cost he/she has to pay to reach the destination which is available in railways. So we considered Rs. 1.5 per minute and calculated the cost of travel based on the time of travel in the bus.

We have considered 10 popular cities of Punjab which are-AMRITSAR, BATHINDA, CHANDIGARH, JALANDHAR, LUDHIANA, PATIALA, PATHANKOT, SANGRUR, MOGA, FEROZEPUR.

Fig: 1. A map of Punjab

Analysis of problem w.r.t. Data structure and Algorithmic technique used

Analysing the problem was quite tough, and implementing it with data structures and algorithms took quite a time as there were so many buses from one particular to one particular destination. We used a new type of data structure called **pair**. By using this data structure we are allocated a memory chunk in which there are two parts from one source to destination-

    a.) The starting time of a particular bus.
    b.) The reaching time of that particular bus.

Adding data in this keyword is done by using make_pair() function.

Now all the inputs which we got from the Punjab roadways site of Chandigarh in this pair keyword, we have created an array num[a][b] that stores the number of bus from a to b.

Now the main part- how to find the minimum path from one city to another?? This is done by the simple Dijkstra algorithm, but it is not the simple Dijkstra algorithm that we studied in class. It's like a developed model of that algorithm that first finds the minimum next city from the starting and then continues to do so till we reach the destination. We have used **array** to the maximum knowledge we could and we have used the basic logic of **graphs**.

Sometimes it was like there is a direct bus from a to b, so our code also tells the time and cost of the direct bus and from any other possible path if it exists. This was quite tricky as each route had it separate and unique time and calculating the cost at the same time was tough. This is also implemented with **pair** and **array**.

The algorithm we developed is shown in the next chapter of **Methodology.**

Methodology used form of pseudo code/flowchart/algorithm

Main data type- pair<pair<int,int>,pair<int,int> > data[10][10][20];

How to add input in this- data[0][1][0]=make_pair(make_pair(7,11),make_pair(11,41));

This means that bus from 0(AMRITSAR) to 1(BATHINDA) leaves Amritsar at 7:11 am and reaches Bathinda at 11:41 am. The last [0] means it's the first from Amritsar to Bathinda.

How number of buses are stored-

int num[10][10];

num[0][1]=3;

This **num[][]** 2-D array means that from 0(AMRITSAR) to 1(BATHINDA) number of buses are=3.

Then the following inputs are taken from the user-

cout<<"ENTER SOURCE AND DESTINATION=";

int a,b;

cin>>a;

cin>>b;

cout<<"ENTER STARTING TIME IN HH MM=";

int h,m;

cin>>h>>m;

Then the algo finds the next city with minimum time. Reach[] is and array that stores 1 if that particular node in the graph is visited or not.

```
miin=-1;
    for(int i=0;i<10;i++){
      if(miin==-1){
        if(reach[i]==0){
          miin=i;
        }
      }
      else{
```

5

```
        if(reach[i]==0)
            { if(time[i].first<time[miin].fir
            st)
                miin=i;
            else if(time[i].first==time[miin].first&&time[i].second<time[miin].second)
                miin=i;
            }
        }
    }
    reach[miin]=1; //make the reach of that node=1.
```

Now this loop calculates the minimum time

```
for(int i=0;i<10;i++){
        int x=-1;          int f1=-1;
        if(i!=miin){
                    for(int j=0;j<num[miin][i];j++){
                        if(x==-1){
                            if(data[miin][i][j].first.first>time[miin].first)
                                x=j;
    else if(data[miin][i][j].first.first==time[miin].first&&data[miin][i][j].first.second>time[miin].second)
                x=j;
            }
            else{
                if(data[miin][i][j].first.first>time[miin].first)
                { if(data[miin][i][j].first.first<time[i].first)
                    x=j;
    else if(data[miin][i][j].first.first==time[i].first&&data[miin][i][j].first.second<time[i].second)
                x=j;
                }
    else if(data[miin][i][j].first.first==time[miin].first&&data[miin][i][j].first.second>time[miin].second)
                { if(data[miin][i][j].first.first<time[i].first)
```

```
                            x=j;
    else if(data[miin][i][j].first.first==time[i].first&&data[miin][i][j].first.second<time[i].second)

                            x=j;

                        }

                    }

                if(x!=-1){ if(time[i].first>data[miin][i][x].second.first)

                    {

                        time[i]=make_pair(data[miin][i][x].second.first,data[miin][i][x].second.second);

                      parent[i]=miin;

                        f1=x;

                    }

 else if(time[i].first==data[miin][i][x].second.first&&time[i].second>data[miin][i][x].second.second)

                    { time[i]=make_pair(data[miin][i][x].second.first,data[miin][i][x].second.second);

                      parent[i]=miin;

                        f1=x;

                    }

                }

            }

        if(f1!=-1){

total[i]=total[miin]+(td(make_pair(data[miin][i][f1].first.first,data[miin][i][f1].first.second),make_pair(da
ta[miin][i][f1].second.first,data[miin][i][f1].second.second)));

            bus[miin][i]=f1;

        }

    }

}
```

Then an array is made that stores the route

```
int p=0;
    int q=b;
    int par[10];
    while(q!=a){
        par[p]=q;
        p++;
        q=parent[q];
    }
    par[p]=a;
```

The same algo is used to calculate time for direct bus.

The following function return the time difference td and then cost=1.5*td is calculated and printed.

```
int td(pair<int,int> i,pair<int,int> f){
    int t;
    if(f.second>i.second){
        t=(f.first-i.first)*60+f.second-i.second;
    }
    else{
        t=(f.first-i.first-1)*60+60-(i.second-f.second);
    }
    return t;
}
```

Results and Conclusion including time and space complexity.
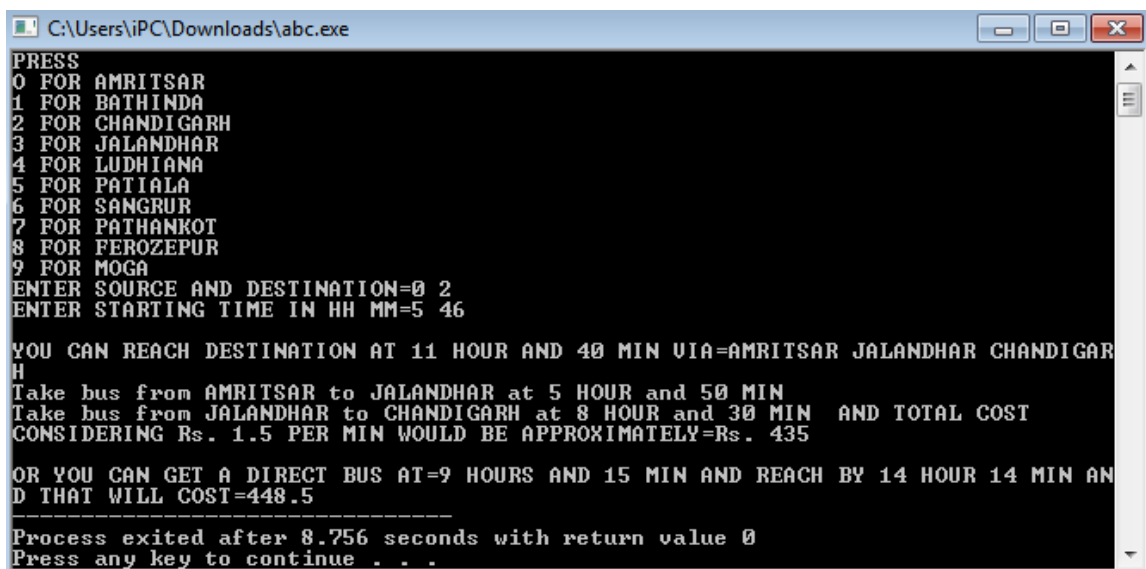
Time complexity- $O(e * n^2)$

Where n- no of cities.

e- no of buses from source to destination.

Space complexity- $O(n^2)$

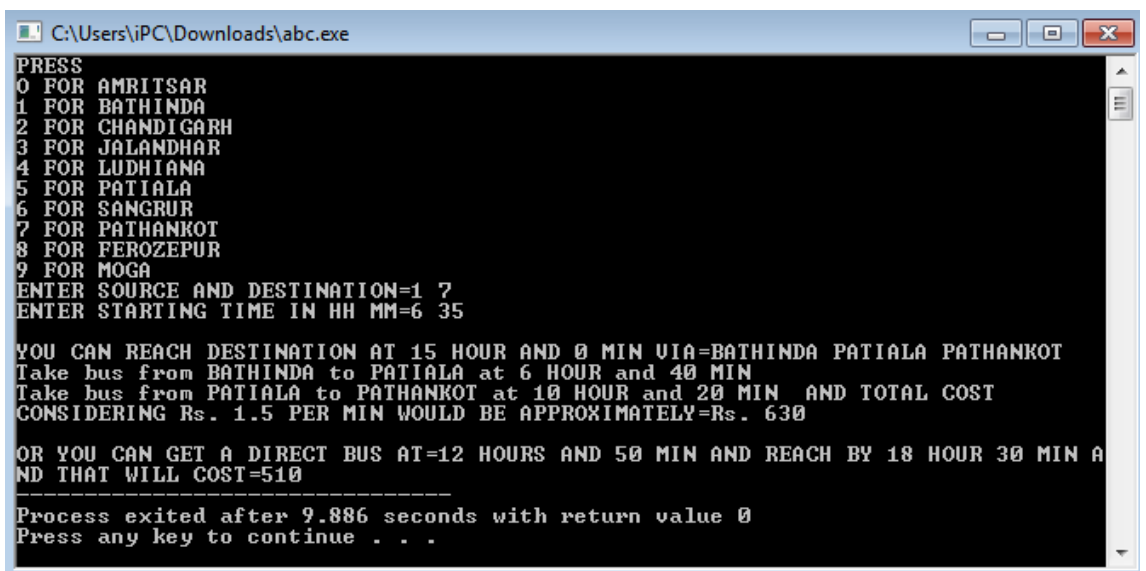Sample outputs-

a.) AMRITSAR TO CHANDIGARH AT 5:46 AM


```
C:\Users\iPC\Downloads\abc.exe

PRESS
0 FOR AMRITSAR
1 FOR BATHINDA
2 FOR CHANDIGARH
3 FOR JALANDHAR
4 FOR LUDHIANA
5 FOR PATIALA
6 FOR SANGRUR
7 FOR PATHANKOT
8 FOR FEROZEPUR
9 FOR MOGA
ENTER SOURCE AND DESTINATION=0 2
ENTER STARTING TIME IN HH MM=5 46

YOU CAN REACH DESTINATION AT 11 HOUR AND 40 MIN VIA=AMRITSAR JALANDHAR CHANDIGAR
H
Take bus from AMRITSAR to JALANDHAR at 5 HOUR and 50 MIN
Take bus from JALANDHAR to CHANDIGARH at 8 HOUR and 30 MIN  AND TOTAL COST
CONSIDERING Rs. 1.5 PER MIN WOULD BE APPROXIMATELY=Rs. 435

OR YOU CAN GET A DIRECT BUS AT=9 HOURS AND 15 MIN AND REACH BY 14 HOUR 14 MIN AN
D THAT WILL COST=448.5
------------------------------------
Process exited after 8.756 seconds with return value 0
Press any key to continue . . .
```

b.) BATHINDA TO PATHANKOT AT 6:35 AM


```
C:\Users\iPC\Downloads\abc.exe

PRESS
0 FOR AMRITSAR
1 FOR BATHINDA
2 FOR CHANDIGARH
3 FOR JALANDHAR
4 FOR LUDHIANA
5 FOR PATIALA
6 FOR SANGRUR
7 FOR PATHANKOT
8 FOR FEROZEPUR
9 FOR MOGA
ENTER SOURCE AND DESTINATION=1 7
ENTER STARTING TIME IN HH MM=6 35

YOU CAN REACH DESTINATION AT 15 HOUR AND 0 MIN VIA=BATHINDA PATIALA PATHANKOT
Take bus from BATHINDA to PATIALA at 6 HOUR and 40 MIN
Take bus from PATIALA to PATHANKOT at 10 HOUR and 20 MIN  AND TOTAL COST
CONSIDERING Rs. 1.5 PER MIN WOULD BE APPROXIMATELY=Rs. 630

OR YOU CAN GET A DIRECT BUS AT=12 HOURS AND 50 MIN AND REACH BY 18 HOUR 30 MIN A
ND THAT WILL COST=510
------------------------------------
Process exited after 9.886 seconds with return value 0
Press any key to continue . . .
```

c.) AMRITSAR TO SANGRUR AT 12:00 NOON

```
C:\Users\iPC\Downloads\abc.exe                                    [ - ][ □ ][ x ]
PRESS
0 FOR AMRITSAR
1 FOR BATHINDA
2 FOR CHANDIGARH
3 FOR JALANDHAR
4 FOR LUDHIANA
5 FOR PATIALA
6 FOR SANGRUR
7 FOR PATHANKOT
8 FOR FEROZEPUR
9 FOR MOGA
ENTER SOURCE AND DESTINATION=0 6
ENTER STARTING TIME IN HH MM=12 00

YOU CAN REACH DESTINATION AT 20 HOUR AND 30 MIN VIA=AMRITSAR JALANDHAR SANGRUR
Take bus from AMRITSAR to JALANDHAR at 14 HOUR and 25 MIN
Take bus from JALANDHAR to SANGRUR at 18 HOUR and 0 MIN   AND TOTAL COST
CONSIDERING Rs. 1.5 PER MIN WOULD BE APPROXIMATELY=Rs. 382.5

NO DIRECT BUS AVAILABLE


------------------------------------
Process exited after 6.618 seconds with return value 0
Press any key to continue . . .
```

d.) PATHANKOT TO MOGA AT 3:00

```
C:\Users\iPC\Downloads\abc.exe                                    [ - ][ □ ][ x ]
0 FOR AMRITSAR
1 FOR BATHINDA
2 FOR CHANDIGARH
3 FOR JALANDHAR
4 FOR LUDHIANA
5 FOR PATIALA
6 FOR SANGRUR
7 FOR PATHANKOT
8 FOR FEROZEPUR
9 FOR MOGA
ENTER SOURCE AND DESTINATION=7 9
ENTER STARTING TIME IN HH MM=3 00

YOU CAN REACH DESTINATION AT 14 HOUR AND 0 MIN VIA=PATHANKOT AMRITSAR BATHINDA M
OGA
Take bus from PATHANKOT to AMRITSAR at 4 HOUR and 0 MIN
Take bus from AMRITSAR to BATHINDA at 7 HOUR and 11 MIN
Take bus from BATHINDA to MOGA at 12 HOUR and 10 MIN   AND TOTAL COST
CONSIDERING Rs. 1.5 PER MIN WOULD BE APPROXIMATELY=Rs. 825

NO DIRECT BUS AVAILABLE


------------------------------------
Process exited after 7.75 seconds with return value 0
Press any key to continue . . .
```

REFERENCES

1.) www.wikipedia.com

2.) www.geeksforgeeks.com

3.) www.youtube.com

4.) www.codechef.com

5.) Advanced data structures- Sartaj Sahni.

6.) Data structures and algorithms-Thomas H.Coreman.

                                    -Charles E. Leiserson.

                                    -Ronald l. Rivest.

                                    -Clifford Stein.