DEPARTMENT OF

COMPUTER SCIENCE ENGINEERING

VASAVI COLLEGE OF ENGINEERING

AUTONOMOUS

IBRAHIMBAGH, HYDERABAD

A PROJECT ON

TRAVEL AND TOURISM

(COURSE- DESIGN AND ANALYSIS OF ALGORITHMS LAB)

BACHELER OF ENGINEERING

IN

COMPUTER SCIENCE

BY

B.SHRUTHI(1602-18-733-104)

P.VISHWACHAND(1602-18-733-120)

TABLE OF CONTENTS

PROJECT ABSTRACT

TRAVEL AND TOURISM

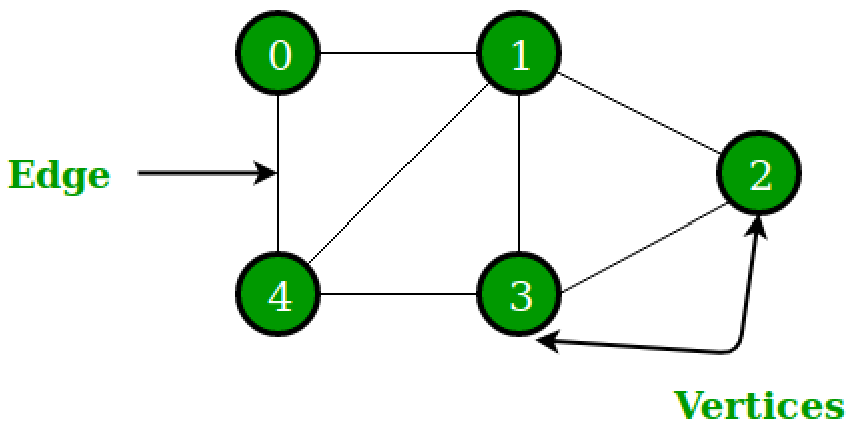
In the world where everyone prefers affordable and fast means of transportation, online platforms which help in booking vehicles have gained momentum off late. Google maps provide a lot of services today that help in comfortable commute

Our project deals with booking system which allows to book to travel different countries which are mentioned in our list. Firstly, the user is prompted to choose an option whether he wants to travel all the countries which are mentioned in out list or move from one country to other country. For this we used three algorithms namely shortest path algorithm (Dijkstra's), Haffman coding and travelling sales man problem. Dijkstra's algorithm is used to find shortest path between two countries which are mentioned by the user. Travelling salesman algorithm is used to travel across all countries in shortest distance. Haffman codes are used to save personal information of the users which provides security. Our platform charges on the basis of distance travelled by a user. After user selects his choice total amount will be displayed and he will be asked whether to confirm his booking or not. As he confirms he will receive an email with time and date of the trip.

DESCRIPTION

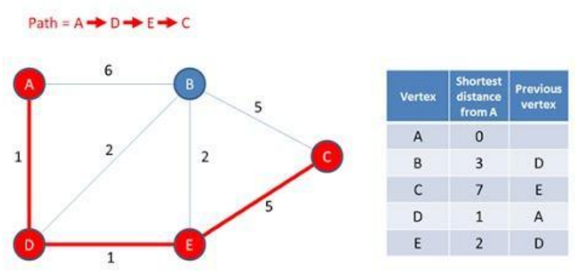
Our project deals with online booking system. We have considered 5 countries in Our project which are connected by airlines. These countries are America, Japan, China, India, Italy, France, Germany, Turkey, Spain and Mexico. The network is stored in the form of a graph.

A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph.



The user is first displayed with the names of the places which are part of the road network. He is prompted to enter the pick up and drop location. After making the appropriate selection the user is asked to enter the number of passengers who want to avail the services. If the number of passengers is more than 1 then the user is provided the option for selecting either cab/auto. If not, the user has three choices. They are->cab, auto and bike. Then, the shortest path is calculated by using all pairs shortest path algorithm. Also, we provide cars for rent and based on the number of hours the cost is printed.

[redacted]



Vertex	Shortest distance from A	Previous vertex
A	0	
B	3	D
C	7	E
D	1	A
E	2	D

[redacted]

This is an algorithm with time complexity O(n^3) and it uses the concept of dynamic programming. The total fare is displayed based on the cost of each edge. The dijsktras algorithm is used to calculate the minimum time required for the vehicle to arrive so that the user can stay informed. Dijkstras algorithm makes use of greedy approach and the time complexity of the algorithm is O(n^2). This is also called as single source shortest path algorithm. The total fare for auto is half the rate of the total fare of the cab. The total fare of a bike is 1/3rd of the fare of the cab. In the end, the OTP which is a 4-digit number is generated and the booking is confirmed.

Data Structures and other concepts used are->

1. Graphs

2. Dynamic programming

3. Greedy approach

4. Random function in C language

5.Libraries used are stdio.h,time.h,stdlib.h,string.h

CODE

```
#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#define SIZE 26

struct register1

{

   char name[30];

   char email[40];

   char password[30];


};

#define MAX_TREE_HT 100

struct MinHeapNode {

   char data;

   unsigned freq;

   struct MinHeapNode *left, *right;

};

struct MinHeap {

   unsigned size;

   unsigned capacity;

   struct MinHeapNode** array;

};

struct MinHeapNode* newNode(char data, unsigned freq)

{

   struct MinHeapNode* temp

      = (struct MinHeapNode*)malloc
```

```
(sizeof(struct MinHeapNode));


    temp->left = temp->right = NULL;
    temp->data = data;
    temp->freq = freq;


    return temp;
}
struct MinHeap* createMinHeap(unsigned capacity)
{


    struct MinHeap* minHeap
        = (struct MinHeap*)malloc(sizeof(struct MinHeap));
    minHeap->size = 0;
    minHeap->capacity = capacity;
    minHeap->array
        = (struct MinHeapNode**)malloc(minHeap->
capacity * sizeof(struct MinHeapNode*));
    return minHeap;
}
void swapMinHeapNode(struct MinHeapNode** a,
                     struct MinHeapNode** b)


{


    struct MinHeapNode* t = *a;
    *a = *b;
    *b = t;
}
void minHeapify(struct MinHeap* minHeap, int idx)

{

    int smallest = idx;
    int left = 2 * idx + 1;
    int right = 2 * idx + 2;

    if (left < minHeap->size && minHeap->array[left]->
freq < minHeap->array[smallest]->freq)
        smallest = left;

    if (right < minHeap->size && minHeap->array[right]->
freq < minHeap->array[smallest]->freq)
        smallest = right;

    if (smallest != idx) {
        swapMinHeapNode(&minHeap->array[smallest],
                        &minHeap->array[idx]);
        minHeapify(minHeap, smallest);
    }
}
int isSizeOne(struct MinHeap* minHeap)
{
   return (minHeap->size == 1);
}
struct MinHeapNode* extractMin(struct MinHeap* minHeap)

{

   struct MinHeapNode* temp = minHeap->array[0];
   minHeap->array[0]
      = minHeap->array[minHeap->size - 1];

   --minHeap->size;
   minHeapify(minHeap, 0);

   return temp;
}
void insertMinHeap(struct MinHeap* minHeap,
               struct MinHeapNode* minHeapNode)

{

   ++minHeap->size;
   int i = minHeap->size - 1;

   while (i && minHeapNode->freq < minHeap->array[(i - 1) / 2]->freq) {

      minHeap->array[i] = minHeap->array[(i - 1) / 2];
      i = (i - 1) / 2;
```

```
    }


    minHeap->array[i] = minHeapNode;
}
void buildMinHeap(struct MinHeap* minHeap)


{


    int n = minHeap->size - 1;
    int i;


    for (i = (n - 1) / 2; i >= 0; --i)
        minHeapify(minHeap, i);
}
void printArr(int arr[], int n)
{
    int i;
    FILE *outfile2;
    outfile2=fopen("huffcode","w");
    for (i = 0; i < n; ++i)


        fprintf(outfile2,"%d",arr[i]);


}
int isLeaf(struct MinHeapNode* root)


{
    return !(root->left) && !(root->right);
}
struct MinHeap* createAndBuildMinHeap(char data[], int freq[], int size)
{

    struct MinHeap* minHeap = createMinHeap(size);

    for (int i = 0; i < size; ++i)
        minHeap->array[i] = newNode(data[i], freq[i]);

    minHeap->size = size;
    buildMinHeap(minHeap);

    return minHeap;
}
struct MinHeapNode* buildHuffmanTree(char data[], int freq[], int size)

{
    struct MinHeapNode *left, *right, *top;
    struct MinHeap* minHeap = createAndBuildMinHeap(data, freq, size);
    while (!isSizeOne(minHeap)) {
        left = extractMin(minHeap);
        right = extractMin(minHeap);
        top = newNode('$', left->freq + right->freq);

        top->left = left;
        top->right = right;

        insertMinHeap(minHeap, top);
    }
    return extractMin(minHeap);
}
void printCodes(struct MinHeapNode* root, int arr[], int top)

{
    if (root->left) {

        arr[top] = 0;
        printCodes(root->left, arr, top + 1);
    }
    if (root->right) {

        arr[top] = 1;
        printCodes(root->right, arr, top + 1);
    }
    if (isLeaf(root)) {
        printArr(arr, top);
    }
}

void HuffmanCodes(char data[], int freq[], int size)

{
    struct MinHeapNode* root

        = buildHuffmanTree(data, freq, size);




    int arr[MAX_TREE_HT], top = 0;


    printCodes(root, arr, top);

}

void freq(char string[]){

    int i, j, length = strlen(string);

    int freq[length];

    for(i = 0; i < strlen(string); i++) {

        freq[i] = 1;

        for(j = i+1; j < strlen(string); j++) {

            if(string[i] == string[j]) {

                freq[i]++;

                string[j] = '\0';

            }

        }




    }


    HuffmanCodes(string,freq,strlen(string));

}

int dijkstra(long int cost[10][10],int startnode,int dst)

{
    int v=dst;
int distance[10],pred[10];
int visited[10],count,mindistance,nextnode,i,j,sum=0,pay[10][10];
int INF=99999;
for(i=0;i<10;i++)
{
for(j=0;j<10;j++)
{
pay[i][j]=cost[i][j];
}
}
for(i=0;i<10;i++)
{
distance[i]=pay[startnode][i];
pred[i]=startnode;
visited[i]=0;
}
distance[startnode]=0;
visited[startnode]=1;
count=1;
while(count<9)
{
mindistance=INF;
for(i=0;i<10;i++)
if(distance[i]<mindistance&&!visited[i])
{
mindistance=distance[i];
```

```
nextnode=i;

}

visited[nextnode]=1;

for(i=0;i<10;i++)

if(!visited[i])

if(mindistance+pay[nextnode][i]<distance[i])

{

distance[i]=mindistance+pay[nextnode][i];

pred[i]=nextnode;

}

count++;

}


for(i=0;i<10;i++)

if(i!=startnode)

{

if(v==i){

   sum+=distance[i];

printf("\nountries u visit in order is\n");

switch(i)

{

  case 0:{

     printf("<-usa");

     break;}

  case 1:{

     printf("<-mexico");

     break;}
  case 2:{
     printf("<-spain");
     break;}
  case 3:{
     printf("<-france");
     break;}
  case 4:{
      printf("<-germany");
     break;}
  case 5:{
      printf("<-italy");
     break;}
  case 6:{
      printf("<-turkey");
     break;}
  case 7:{
      printf("<-india");
     break;}
  case 8:{
      printf("<-china");
     break;}
  case 9:{
      printf("<-japan");
     break;}


}
j=i;
```

```
do
{
j=pred[j];
sum+=distance[i];
switch(j)
{
  case 0:{
      printf("<-usa");
      break;}
  case 1:{
      printf("<-mexico");
      break;}
  case 2:{
      printf("<-spain");
      break;}
  case 3:{
      printf("<-france");
      break;}
  case 4:{
       printf("<-germany");
      break;}
  case 5:{
       printf("<-italy");
      break;}
  case 6:{
       printf("<-turkey");
      break;}
  case 7:{
       printf("<-india");
     break;}
  case 8:{
       printf("<-china");
     break;}
  case 9:{
       printf("<-japan");
     break;}
}
}while(j!=startnode);
}}
return sum;
            }
long int travelling(long int graph[10][10],char sourse[15])
{
    if(strcmp("USA",sourse)==0)
        return 38620;
    else if(strcmp("MEXICO",sourse)==0)
        return 40250;
    else if(strcmp("SPAIN",sourse)==0)
        return 42300;
    else if(strcmp("FRANCE",sourse)==0)
        return 39580;
    else if(strcmp("GERMANY",sourse)==0)
        return 45030;
    else if(strcmp("ITALY",sourse)==0)
```

```
        return 43230;
    else if(strcmp("TURKEY",sourse)==0)
        return 41720;
    else if(strcmp("INDIA",sourse)==0)
        return 43240;
    else if(strcmp("CHINA",sourse)==0)
        return 41230;
    else if(strcmp("USA",sourse)==0)
        return 52340;
    else
    {
        printf("                                        invalid input\n");
        exit(1);
    }
}
int main()
{
    int key;
    char choice[4];
    int choice1;
    struct register1 person[20];
    int res[256];
    FILE *outfile1,*outfile2;
    outfile1=fopen("actual","w");
    int i=0,j;
    printf("                              ");
    printf("***************");
    printf("                              \n");
    printf("                              ");
    printf("WELCOME TO TRAVEL AND TOURISM\n");
    printf("\n");
    printf("                              ");
    printf("Press any key to continue\n");
    scanf("%d",&key);
    if(key!=-1){
    printf("                              ");
    printf("We are happy that you are heading forward with us\n");
    printf("              ");
    printf("Please register into our site to get 10 percent off on your first 5 travels\n");
    printf("               ");
    printf("If you are not interested to register and continue then press NO else YES\n");
    printf("\n");
    scanf("%s",choice);
    char name[15],email[15],password[15];
    if(strcmp("YES",choice)==0)
    {
        int retfreq[20];
        printf("enter your name\n");
        scanf("%s",name);
        printf("enter your email address\n");
        scanf("%s",email);
        printf("enter your password\n");
        scanf("%s",password);
        fprintf(outfile1,"%s",name);
```

```
    fprintf(outfile1,"%s",email);

    fprintf(outfile1,"%s",password);

    freq(name);

}

printf("                                    \n");

printf(" Countries which are included in our plan are\n");

    printf("                                        ");

    printf("0:USA\n");

    printf("                                        ");

    printf("1:MEXICO\n");

    printf("                                        ");

    printf("2:SPAIN\n");

    printf("                                        ");

    printf("3:FRANCE\n");

    printf("                                        ");

    printf("4:GERMANY\n");

    printf("                                        ");

    printf("5:ITALY\n");

    printf("                                        ");

    printf("6:TURKEY\n");

    printf("                                        ");

    printf("7:INDIA\n");

    printf("                                        ");

    printf("8:CHINA\n");

    printf("                                        ");

    printf("9:JAPAN\n");

printf("                                              \n");
printf("please enter your choice\n");
printf("                              ");
printf("1:visit all countries(world tour)\n");
printf("                              ");
printf("2:source and destination trip\n");
scanf("%d",&choice1);
char source[15],destination[15];
long int graph[10][10]={{0,1893,4713,0,7767,0,0,13568,0,10144},{1893,0,9031,0,0,0,0,0,0,0},{4713,9031,0,826,0,0,0,0,0,0},{0,0,826,0,872,929,0,0,0,0},{7767,0,0,872,0,0,0,0,0,9437},{0,0,0,929,0,0,1905,0,0,0},{0,0,0,0,0,1905,0,4566,0,0},{13568,0,0,0,0,0,4566,0,2910,0},{10144,0,0,0,0,0,0,0,3137,0}};
double distance;
int choice3;
char account[20];
int s,d;
switch(choice1)
{
case 1:
  printf("enter a place from where u want to start your journey \n");
  scanf("%s",source);
  distance=travelling(graph,source);
  printf("                         ");
  printf("Total amount u have to pay is %f",distance*10);
  printf("\n                         ");
  printf("please press 1 to continue\n");
  scanf("%d",&choice3);
  if(choice3==1)
```

```
    {
      printf("                              ");
      printf("enter your account number to confirm your request and pay\n");
      scanf("%s",account);
      freq(account);
      printf("                           \n");
        printf("your payment is successful and u will get a confirmation mail and date
for your travel");



    }
    break;
  case 2:
    printf("                              ");
    printf("enter your source and destination according to the table displayed\n");
    scanf("%d %d",&s,&d);
    distance=dijkstra(graph,s,d);
    printf("\nTotal amount u have to pay is %f",distance*10);
    printf("\n                              ");
    printf("please press 1 to continue\n");

    scanf("%d",&choice3);
    if(choice3==1)
    {
      printf("                              ");
      printf("enter your account number to confirm your request and pay\n");
      scanf("%s",account);
```

```
          freq(account);

          printf("                              \n");

           printf("\nyour payment is successful and u will get a confirmation mail and date
for your travel");



       }

   }

   }

   else{

       exit(1);

   }

return 0;

}
```

OUTPUT

```
***************

                                   WELCOME TO TRAVEL AND TOURISM


                                   Press any key to continue

1

                                   We are happy that you are heading forward with us

              Please register into our site to get 10 percent off on your first 5 travels

               If you are not interested to register and continue then press NO else YES


YES

enter your name

shruthi
```

```
enter your email address

shruthiapr3@gmail.com

enter your password

bheemreddy


 Countries which are included in our plan are

                                    0:USA

                                    1:MEXICO

                                    2:SPAIN

                                    3:FRANCE

                                    4:GERMANY

                                    5:ITALY

                                    6:TURKEY

                                    7:INDIA

                                    8:CHINA

                                    9:JAPAN


please enter your choice

                                           1:visit all countries(world tour)

                                           2:source and destination trip

1

enter a place from where u want to start your journey

INDIA

                                  Total amount u have to pay is 432400.000000

                                 please press 1 to continue

1

                                enter your account number to confirm your request and pay
```

```
1234


your payment is successful and u will get a confirmation mail and date for your travel
```

TESTCASES

```
***************
                              WELCOME TO TRAVEL AND TOURISM


                              Press any key to continue
1
                              We are happy that you are heading forward with us
            Please register into our site to get 10 percent off on your first 5 travels
             If you are not interested to register and continue then press NO else YES


NO


 Countries which are included in our plan are
                                0:USA
                                1:MEXICO
                                2:SPAIN
                                3:FRANCE
                                4:GERMANY
                                5:ITALY
                                6:TURKEY
                                7:INDIA
                                8:CHINA
                                9:JAPAN
```

```
please enter your choice

                                        1:visit all countries(world tour)

                                        2:source and destination trip

2

                               enter your source and destination according to the table displayed

0 4


ountries u visit in order is

<-germany<-mexico<-france<-usa

Total amount u have to pay is 134250.000000

                              please press 1 to continue

1

                             enter your account number to confirm your request and pay

1234



your payment is successful and u will get a confirmation mail and date for your travel
```

REFERENCES

- https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/
- TEXTBOOK REFERRED- DESIGN AND ANALYSIS OF ALGORITHMS BY ELLIS HOROWITZ AND SATRAJ SAHANI