Advanced Data Structures Assignment

Name : Shrirang R Mhalgi

Roll number : 222006

Class : SE B

Batch : B1

--------------------------------------------------------------------------------------------------------------------------------------

**Problem Statement:**

There are flight paths between cities. If there is a flight between city A and city B then there is an edge between the cities. The cost of the edge can be the time that flight takes to reach city B from A, or the amount of fuel used for the journey. Represent this as a graph. The node can be represented by airport name or name of the city. Use adjacency list representation of the graph or use adjacency matrix representation of the graph. Justify the storage representation used.

**Code:**

#include<iostream>

using namespace std;

int city\_num,i,j;

char ans,t;

class node{

string data;

char time;

node \*next;

friend class Graph;

};

class Graph{

string city[10];

node \*arr[10];

public:

int create();

char city\_matrix[10][10];

int matrix();

void dis\_m();

int list();

void dis\_l();

};

int Graph :: create(){

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

cin>>city\_num;

for(i=0;i<city\_num;i++)

{

cout<<"City "<<i+1<<" : ";

cin>>city[i];

}

return city\_num;

}

int Graph :: matrix(){

for(i=0;i<city\_num;i++)

{

for(j=0;j<city\_num;j++)

{

city\_matrix[i][j]='-';

}

}

for(i=0;i<city\_num;i++)

{

for(j=0;j<city\_num;j++)

{

if(i<=j)

continue;

else

{

cout<<"Is there flight between "<<city[i]<<" and "<< city[j]<<" (y/n)"<<endl;

cin>>ans;

if(ans=='y'||ans=='Y')

{

cout<<"Enter the time taken to fly : ";

cin>>t;

city\_matrix[i][j]=t;

city\_matrix[j][i]=t;

}

}

}

}

}

void Graph :: dis\_m(){

cout<<"\n------------------ADJACENCY MATRIX-------------------\n\n";

for(i=0;i<city\_num;i++)

{

cout<<"\t"<<city[i];

}

cout<<endl;

for(i=0;i<city\_num;i++)

{

cout<<endl<<city[i]<<"\t";

for(j=0;j<city\_num;j++)

{

cout<<city\_matrix[i][j]<<"\t";

}

}

}

int Graph :: list(){

node \*New,\*temp1,\*temp2,\*t1,\*t2;

for(i=0;i<city\_num;i++)

{

New = new node;

New->next=NULL;

New->time='-';

New->data=city[i];

arr[i]=New;

}

for(i=0;i<city\_num;i++){

for(j=i+1;j<city\_num;j++){

cout<<"Is there flight between "<<city[i]<<" and "<< city[j]<<" (y/n)"<<endl;

cin>>ans;

if(ans=='y'||ans=='Y')

{

temp1 = new node;

temp1->next = NULL;

temp2 = new node;

temp2->next = NULL;

temp1->data = city[j];

temp2->data = city[i];

cout<<"Enter the time taken to fly : ";

cin>>t;

temp1->time=t;

temp2->time=t;

t1 = arr[i];

while(t1->next!=NULL)

t1 = t1->next;

t1->next = temp1;

t2=arr[j];

while(t2->next!=NULL)

t2=t2->next;

t2->next = temp2;

}

else

continue;

}

}

}

void Graph :: dis\_l(){

cout<<endl<<"----------------ADJACENCY LIST---------------\n";

node \*curr;

i=0;

while(i<city\_num){

curr = arr[i];

cout<<curr->data;

curr=curr->next;

while(curr!=NULL)

{

cout<<" -> "<<curr->data<<","<<curr->time;

curr=curr->next;

}

cout<<"\n";

i++;

}

}

int main(){

Graph ob;

ob.create();

int choice;

do{

cout<<"\n\n------------------ MENU -------------------";

cout<<"\n1 : Adjacency matrix";

cout<<"\n2 : Adjacency list";

cout<<"\n3 : Exit";

cout<<"\nEnter your choice : ";

cin>>choice;

switch(choice){

case 1 : ob.matrix();

ob.dis\_m();

break;

case 2 : ob.list();

ob.dis\_l();

break;

}

}while(choice!=3);

}

**Output:**

Enter the number of cities : 4

City 1 : pune

City 2 : mumbai

City 3 : delhi

City 4 : chennai

------------------ MENU -------------------

1 : Adjacency matrix

2 : Adjacency list

3 : Exit

Enter your choice : 1

Is there flight between mumbai and pune (y/n)

n

Is there flight between delhi and pune (y/n)

y

Enter the time taken to fly : 3

Is there flight between delhi and mumbai (y/n)

y

Enter the time taken to fly : 2

Is there flight between chennai and pune (y/n)

y

Enter the time taken to fly : 4

Is there flight between chennai and mumbai (y/n)

n

Is there flight between chennai and delhi (y/n)

y

Enter the time taken to fly : 5

------------------ADJACENCY MATRIX-------------------

pune mumbai delhi chennai

pune - - 3 4

mumbai - - 2 -

delhi 3 2 - 5

Chennai 4 - 5 -

------------------ MENU -------------------

1 : Adjacency matrix

2 : Adjacency list

3 : Exit

Enter your choice : 2

Is there flight between pune and mumbai (y/n)

n

Is there flight between pune and delhi (y/n)

y

Enter the time taken to fly : 3

Is there flight between pune and chennai (y/n)

y

Enter the time taken to fly : 4

Is there flight between mumbai and delhi (y/n)

n

Is there flight between mumbai and chennai (y/n)

y

Enter the time taken to fly : 5

Is there flight between delhi and chennai (y/n)

n

----------------ADJACENCY LIST---------------

pune -> delhi,3 -> chennai,4

mumbai -> chennai,5

delhi -> pune,3

chennai -> pune,4 -> mumbai,5

------------------ MENU -------------------

1 : Adjacency matrix

2 : Adjacency list

3 : Exit

Enter your choice : 3