201533661 이승수’s Algorithm hoework#7 date: 2016. 10. 12.

<code>

#include <stdio.h>

#include <string.h>

struct trainList{

char currentCity[20];

//int trainNumber;

int depHour;

int depMin;

int arrHour;

int arrMin;

}trainAtStation[100][100];//cityNum,trainNum

FILE \*inF, \*outF;

int cityNum, trainNum;

int visitedCity[100] = {0}; //S

int curHour = 0, curMin = 0; //

int curCity = 0;

int timeTable[100][100] = { 0 }; //from,to

int shortestTrainIndex[100] = {0};

int shortestIndex = 0;

int HI = 0;

void main()

{

char city[100][20] = {"\n"};

int departureHour=0,departureMin=0;

char from[20];

char to[20];

inF = fopen("input#1.txt","r");

fscanf(inF,"%d",&cityNum);

for (int i = 0; i < cityNum; i++)

{

fscanf(inF,"%s", city[i]);

}

fscanf(inF,"%d",&trainNum);

for (int j = 0; j < trainNum; j++)

{

int stationNum;

int trainStarted = 0;

fscanf(inF,"%d",&stationNum);

for (int i = 0; i < stationNum; i++)

{

int depHour = 0, depMin = 0;

char station[20];

fscanf(inF,"%02d%02d",&depHour,&depMin);

fscanf(inF,"%s",station);

for (int k = 0; k < cityNum; k++)

{

if (strcmp(city[k], station) == 0&&trainStarted==0)

{

strcpy(trainAtStation[k][j].currentCity,station);

trainAtStation[k][j].depHour = depHour;

trainAtStation[k][j].depMin = depMin;

trainStarted++;

}

else if (strcmp(city[k], station) == 0)

{

strcpy(trainAtStation[k][j].currentCity, station);

trainAtStation[k][j].arrHour = depHour;

trainAtStation[k][j].arrMin = depMin;

}

}

}

}

fscanf(inF,"%02d%02d",&departureHour,&departureMin);

fscanf(inF,"%s",from);

fscanf(inF,"%s",to);

fclose(inF);

printf("%d\n", HI++);//0

curHour = departureHour;

curMin = departureMin;// time count from departureHour,Min

for (int i = 0; i < cityNum; i++) //

{

if (strcmp(from, city[i]) == 0) //start

curCity = i;

if (strcmp(to, city[i]) == 0) //end

;

}

int curTrainIndex = 0;//

shortestTrainIndex[shortestIndex++] = curTrainIndex;

while (strcmp(city[curCity],to)!=0)//for (int i = curCity; i < cityNum;)

{

int nextTrainIndex = 0; //

int firstDepartTime = 360;

int curDestinationCity = curCity;

for (int k = curCity; k < cityNum;k++) // find train's destination city's index

{

if (Time(trainAtStation[k][curTrainIndex].arrHour, trainAtStation[k][curTrainIndex].arrMin) != 0)

curDestinationCity = k;

} //ok

printf("curDestinationCity:%d\n",curDestinationCity);

curHour = trainAtStation[curDestinationCity][curTrainIndex].arrHour; //arrived at current train's destination

curMin = trainAtStation[curDestinationCity][curTrainIndex].arrMin;

curCity = curDestinationCity;

printf("curTime(%02d:%02d), trainIndex:%d\n",curHour,curMin,curTrainIndex);

for (int j = 0; j < trainNum; j++)//find next train

{

if (Time(trainAtStation[curDestinationCity][curTrainIndex].arrHour, trainAtStation[curDestinationCity][curTrainIndex].arrMin)<=Time(trainAtStation[curDestinationCity][j].depHour, trainAtStation[curDestinationCity][j].depMin)) //&& Time(trainAtStation[curDestinationCity][j].depHour, trainAtStation[curDestinationCity][j].depMin)<firstDepartTime)

{

nextTrainIndex = j;

firstDepartTime = Time(trainAtStation[curDestinationCity][nextTrainIndex].depHour, trainAtStation[curDestinationCity][nextTrainIndex].depMin);

printf(" nextTrainIndex: %d\n",nextTrainIndex);

}

}

curHour = trainAtStation[curDestinationCity][nextTrainIndex].depHour; //after waiting, get on next train

curMin = trainAtStation[curDestinationCity][nextTrainIndex].depMin;

curTrainIndex = nextTrainIndex;

shortestTrainIndex[shortestIndex++] = curTrainIndex;

}

for (int i = 0; i < cityNum; i++)

{

printf("%d ",shortestTrainIndex[i]);

}

printf("shortestINdex:%d",shortestIndex);

outF = fopen("output#1.txt","w");

//fprintf();

fprintf(outF,"Departure %02d%02d %s\n", trainAtStation[0][shortestTrainIndex[0]].depHour, trainAtStation[0][shortestTrainIndex[0]].depMin, trainAtStation[0][shortestTrainIndex[0]].currentCity);

fprintf(outF, "Arrival %02d%02d %s\n", trainAtStation[shortestIndex - 1][shortestTrainIndex[shortestIndex - 2]].arrHour, trainAtStation[shortestIndex - 1][shortestTrainIndex[shortestIndex - 2]].arrMin, trainAtStation[shortestIndex - 1][shortestTrainIndex[shortestIndex - 2]].currentCity);

fclose(outF);

}

int Time(int hour,int minute)

{

return hour \* 60 + minute;

}