پروژه ساختمان داده

استاد:

سمیرا خدابنده لو

هانیه اکبری

۹۹۱۲۳۵۸۰۰۳

شرح پروژه

هدف از انجام این پروژه، استفاده از الگوریتمی مشخص برای پیدا کردن کوتاه ترین مسیر است.

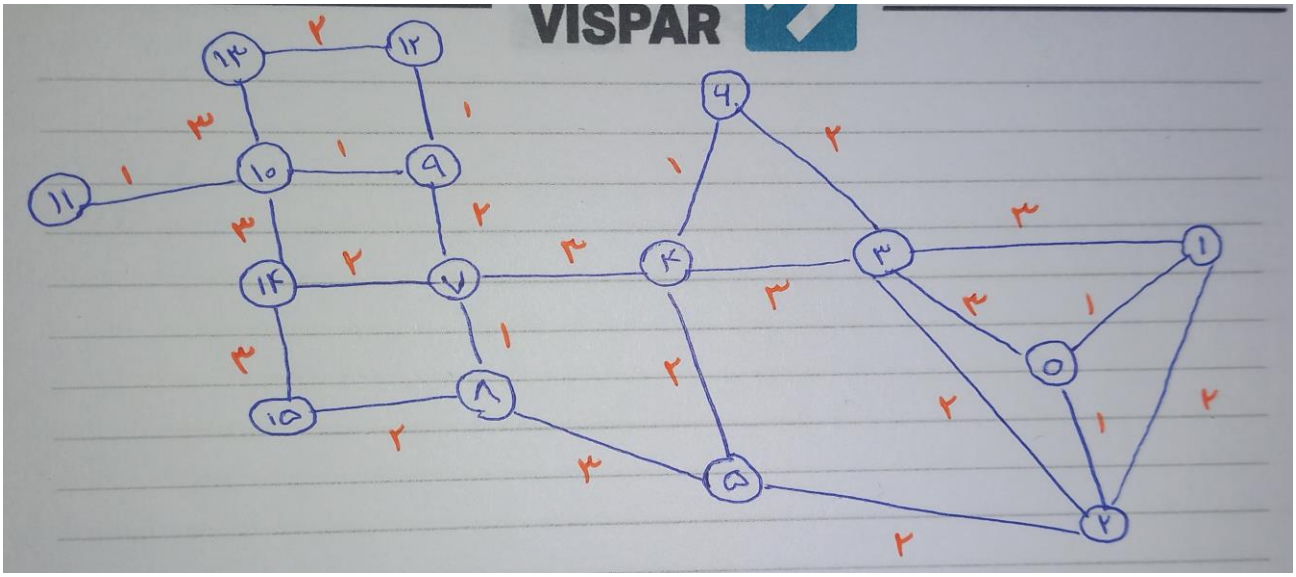
البته مدل سازی نقشه هم یکی از چالش های این پروژه بود.

ساختار برنامه

با استفاده از وکتور، وزن یال ها ذخیره شده است. یک وکتور برای فاصله یال‌ها، و دیگری برای مقدار ترافیک بین یال‌ها.

ساختار کلی به این شکل است که ابتدا کاربر درخواستی را ارائه می دهد (پیدا کردن بهترین مسیر یا کوتاه ترین مسیر) سپس یک شی از کلاس mappath ساخته میشود. و هزینه یا مسافت به همراه مسیر بین گره ها نمایش داده می‌شود.

تبدیل نقشه به گراف



مبدا : ۰

مقصد : ۱۱

ایجاد وکتور دو بعدی

ترافیک:

سطر ۰ ستون ۰ : ۰ (از خود نود به خود نود ترافیکی وجود ندارد.)

سطر ۰ ستون ۱ : ۱ (ترافیک از نود صفر به نود یک برابر با یک است.)

سطر ۰ ستون ۲ : ۱

سطر ۰ ستون ۳ : ۳

سطر ۰ ستون ۴ : ۰ (از نود صفر به نود چهار هیچ یالی وجود ندارد.)

سطر ۰ ستون ۵ : ۰

سطر ۰ ستون ۶ :۰

سطر ۰ ستون ۷ :۰

سطر ۰ ستون ۸ :۰

.

.

سطر ۱ ستون ۰ : ۱

سطر ۱ ستون ۱ : ۰

.

.

سطر ۲ ستون ۰ : ۱

سطر ۲ ستون ۱ : ۲

به همین ترتیب وکتور ترافیک را ایجاد کردیم. (وکتور مسافت هم همینطور تشکیل می‌شود.)

وکتورهای برنامه:

```
//trafic amount in graph
 vector<vector<double>> trafic = {{0, 1, 1, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                                  {1, 0, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                                  {0, 2, 0, 2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                                  {0, 3, 2, 0, 3, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0},
                                  {0, 0, 0, 3, 0, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0},
                                  {0, 0, 2, 0, 2, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0},
                                  {0, 0, 0, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                                  {0, 0, 0, 0, 3, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0},
                                  {0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 2},
                                  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0},
                                  {0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 3, 3},
                                  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0},
                                  {0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 2, 0},
                                  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 2, 0, 0},
                                  {0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 3, 0, 0, 0, 3},
                                  {0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 3, 0}};

//distance amount in graph
vector<vector<double>> dist = {{0, 0.2, 0.2, 0.2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                               {0.2, 0, 1.4, 1.3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                               {0.2, 1.4, 0, 0.2, 0, 2.4, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                               {0.2, 1.3, 0.2, 0, 2.1, 0, 1.9, 0, 0, 0, 0, 0, 0, 0, 0},
                               {0, 0, 0, 2.1, 0, 1.9, 2, 0.1, 0, 0, 0, 0, 0, 0, 0},
                               {0, 0, 2.4, 0, 1.9, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0},
                               {0, 0, 0, 1.9, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                               {0, 0, 0, 0, 0.1, 0, 0, 0, 0.1, 0.1, 0, 0, 0, 3.5, 0},
                               {0, 0, 0, 0, 0, 3, 0, 0.1, 0, 0, 0, 0, 0, 0, 4.1},
                               {0, 0, 0, 0, 0, 0, 0, 0.1, 0, 0, 2, 0, 1.1, 0, 0},
                               {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0.2, 0, 0.6, 0.8, 0},
                               {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.2, 0, 0, 0, 0, 0},
                               {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.1, 0, 0, 0, 1.8, 0, 0},
                               {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.6, 0, 1.8, 0, 0, 0},
                               {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.8, 0, 0, 0, 0, 1.2},
                               {0, 0, 0, 0, 0, 0, 0, 0, 0, 4.1, 0, 0, 0, 0, 0, 1.2, 0}};
```

ساختار main

```
// make an object
mappath snap(dist, trafic);
int command = 1;
while (command < 3 && command > 0)
{
    cout << "************************" << endl;
    cout << "chose one option\n1-Shortestpath\n2-Bestpath\n3-Exit" << endl;
    cout << "************************" << endl;
    cin >> command;
    if (command == 1)
    {
        snap.get_path(SHORTESTPATH);
    }
    else if (command == 2)
    {
        snap.get_path(BESTPATH);
    }
}

return 0;
```

اگر کاربر عدد یک را انتخاب کند، کوتاه‌ترین مسیر نمایش داده می‌شود و اگر عدد ۲ را انتخاب کند بهترین مسیر محاسبه می‌شود و با انتخاب عدد ۳ برنامه به پایان می‌رسد.

ساختار کلاس mappath

C: > Users > hani > project > C mappath.h

```h
#ifndef MAPPATH_H
#define MAPPATH_H
#include <vector>
#include <iostream>
#include <limits.h>
#include <stdio.h>
using namespace std;
enum req
{
    SHORTESTPATH,
    BESTPATH
};

class mappath
{
private:
    vector<vector<double>> traficgraph;   // a vector for store trafic amount
    vector<vector<double>> distancegraph; // a vector for store distance amount
    vector<int> node;                     // a vector for store path node
    double algorithm(int request);        // algorithm for find best path or shortest path in graph
    int minimumdis(vector<double> &, vector<bool> &);

public:
    mappath(vector<vector<double>>, vector<vector<double>>); // constructor
    ~mappath();
    void get_path(int req); // show cost or distance of path
    void path();            // show node path and trafic average
};

#endif
```

❊ برای یافتن کوتاه ترین (کمترین مسافت) و یا بهترین (کمترین مقدار ترافیک مسافت) مسیر از الگوریتم دایجسترا استفاده شده است.

*enum req صرفا برای خوانایی بهتر برنامه است.

کانستراکتور کلاس mappath :

```
mappath::mappath(vector<vector<double>> pathdist, vector<vector<double>>
pathtrafic)
{
    this->distancegraph = pathdist;
    this->traficgraph = pathtrafic;
}
```

این کانستراکتور دو وکتوری که در main برنامه تشکیل شده است را در دو وکتور داخل کلاس کپی می‌کند.

خروجی برنامه

```
hni_stan@DESKTOP-7J3DRQ5:/mnt/c/users/hani/project/cmake$ ./app

************************
chose one option
1-Shortestpath
2-Bestpath
3-Exit
************************
1

kotah tarin masir ba fasele 4.7 az mabda
--------------------------------
node path:
11<==8<==7<==5<==3<==2<==0
average trafic for this path: 0.5

************************
chose one option
1-Shortestpath
2-Bestpath
3-Exit
************************
2

behtarin masir ba hazine 16.5 az mabda
node path:
11<==8<==7<==5<==3<==2<==0
average trafic for this path: 0.5

************************
chose one option
1-Shortestpath
2-Bestpath
3-Exit
************************
3
hni_stan@DESKTOP-7J3DRQ5:/mnt/c/users/hani/project/cmake$
```