

دانشگاه بوعلی سینا

پروژه پایانی ساختمان داده ها

استاد سمیرا خدابنده لو

محمد رضا حیدرنیا – 9912354018

تابستان 1401

شرح پروژه

هدف اصلی این پروژه پیدا کردن کوتاه ترین مسیر از یک مبدا به مقصد مشخص، با توجه به مسافت و ضریب ترافیک در نقشه بود.

ساختار برنامه

ابتدا نقشه را با استفاده از یک گراف مدل سازی کرده و سپس گره ها و وزن یال ها را در یک آرایه ذخیره کردم. دو آرایه در برنامه وجود دارد که یکی مسافت بین گره ها را ذخیره میکند و دیگری مقدار مسافت ضرب در ترافیک.

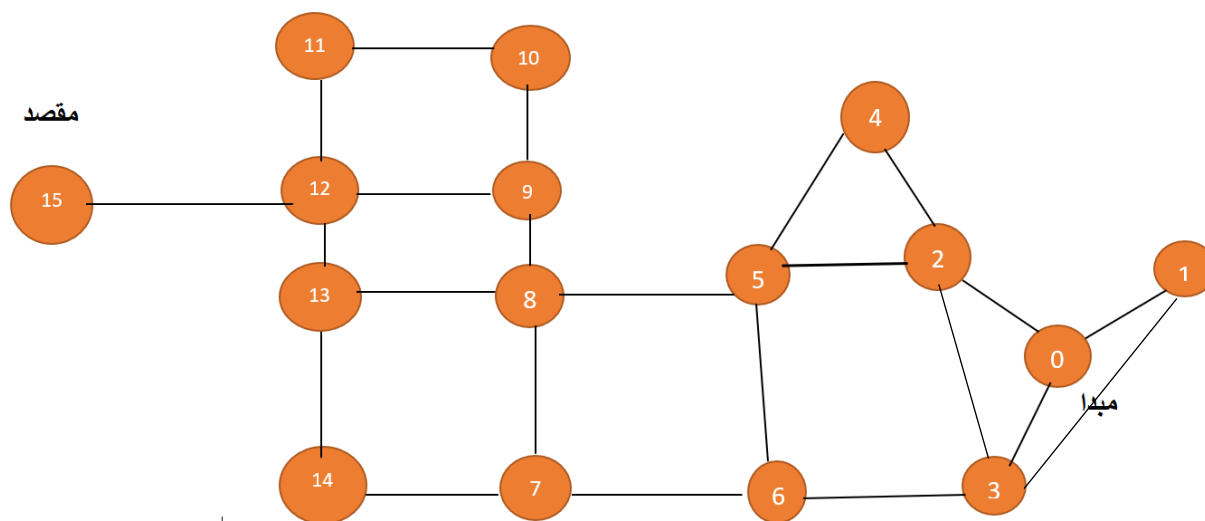
*در پیاده سازی این برنامه از کلاس استفاده شده است.

آرایه های تابع main:

```
float costpath[16][16] = {{0, 0.2, 0.6, 0.2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
                          {0.2, 3.9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
                          {0.6, 3.9, 0, 0, 3.8, 6.3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
                          {0.2, 0, 0, 0, 0, 0, 4.8, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
                          {0, 0, 3.8, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
                          {0, 0, 6.3, 0, 2, 0, 3.8, 0, 0.3, 0, 0, 0, 0, 0, 0, 0 },
                          {0, 0, 0, 4.8, 0, 3.8, 0, 9, 0, 0, 0, 0, 0, 0, 0, 0 },
                          {0, 0, 0, 0, 0, 0, 9, 0, 0.1, 0, 0, 0, 0, 8.2, 0, 0 },
                          {0, 0, 0, 0, 0, 0.3, 0, 0.1, 0, 0.2, 0, 0, 0, 7, 0, 0 },
                          {0, 0, 0, 0, 0, 0, 0, 0, 0.2, 0, 1.1, 0, 2, 0, 0, 0 },
                          {0, 0, 0, 0, 0, 0, 0, 0, 0, 1.1, 0, 3.6, 0, 0, 0, 0 },
                          {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3.6, 0, 1.8, 0, 0, 0 },
                          {0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 1.8, 0, 2.9, 0, 0.2 },
                          {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2.9, 0, 3.6, 0 },
                          {0, 0, 0, 0, 0, 0, 0, 8.2, 0, 0, 0, 0, 0, 0, 3.6, 0, 0 },
                          {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.2, 0, 0 }};

float pathdis[16][16] = {{0, 0.2, 0.2, 0.2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                          {0.2, 0.1.3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                          {0.2, 1.3, 0, 0, 1.9, 2.1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                          {0.2, 0, 0, 0, 0, 0, 2.4, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                          {0, 0, 1.9, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
                          {0, 0, 2.1, 0, 2, 1.9, 0, 0, 0.1, 0, 0, 0, 0, 0, 0, 0},
                          {0, 0, 0, 2.4, 0, 1.9, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0},
                          {0, 0, 0, 0, 0, 0, 3, 0, 0.1, 0, 0, 0, 0, 0, 4.1, 0},
                          {0, 0, 0, 0, 0, 0.1, 0, 0.1, 0, 0.1, 0, 0, 0, 3.5, 0, 0},
                          {0, 0, 0, 0, 0, 0, 0, 0, 0.1, 0, 1.1, 0, 2, 0, 0, 0},
                          {0, 0, 0, 0, 0, 0, 0, 0, 0, 1.1, 0, 1.8, 0, 0, 0, 0},
                          {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.8, 0, 0.6, 0, 0, 0},
                          {0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0.6, 0, 0.8, 0, 0.2},
                          {0, 0, 0, 0, 0, 0, 0, 0, 3.5, 0, 0, 0, 0.8, 0, 1.2, 0},
                          {0, 0, 0, 0, 0, 0, 0, 4.1, 0, 0, 0, 0, 0, 1.2, 0, 0},
                          {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.2}};
```

گراف:



در این گراف گره شماره 15 ، گره مقصد است و باید کوتاه ترین مسیر تا این گره توسط برنامه پیدا شود. در این برنامه بهترین مسیر از نظر هزینه و کوتاه ترین مسیر یافت میشود. برای پیدا کردن کوتاه ترین مسیر و بهترین مسیر از الگوریتم دایجسترا استفاده میکنیم. آرایه های موجود و سائز آرایه را به یک شی از کلاس میدهیم.

ساختار کلاس:

```
#ifndef bestpath_HPP
#define bestpath_HPP
using namespace std;
#include <stdio.h>
#include <iostream>
#include <limits.h>

class bestpath
{
private:
    float** costpath; // best way that include both items trafic and distance
    float** distance; // find distance of best way
    int* store;        // store nodes for find path(store previous node)
    int g_size = 0;
    float dijkstra(int check); //algorithm for find best path
    int minimumdis(float [], bool[]);

public:
    bestpath(float **graph,float **dis, int size); //constructor
    ~bestpath();//distructor
    void showpath(int check);
    int nodepath();
};
```

اگر کاربر بهترین مسیر را درخواست کند آرایه `costpath` به متد `Dijkstra` داده میشود. و اگر کوتاه ترین مسیر را بخواهد آرایه `distance` به این متد داده میشود.

در نهایت هزینه یا مسافت طی شده به همراه نمایش مسیر به کاربر نشان داده میشود.

خروجی برنامه:

```
*****
What do you want?
-----
1-show bestpath
2-show shortest path
3-Exit
*****
1
cost of best path: 9.1
your path is : 0-->2-->4-->5-->8-->9-->12-->15
```

با تشکر - محمدرضا حیدرنیا