桂林电子科技大学

**数据结构与算法**  实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | **实验七 校园导航** | | | | | | | |  | 辅导员意见：  成绩 辅导员  签 名 |
| 院 系 | 计算机与信息安全学院 | | | 专业 | | 计算机大类 | | |
| 学 号 | 1800300636 | | | 姓名 | | 奚锐 | | |
| 实验日期 | 2019 | 年 | 11 | | 月 | | 9 | 日 |
|  |  | | | | | | | |

**一、实验目的**

（1）掌握图的邻接矩阵或邻接表表示方法

（2）理解掌握最短路径Dijkstra算法。

**二、实验具体内容**

**1、实验题目1：**

为校园导航问题抽象数据结构模型、选择存储表示、设计实现最短路径Dijkstra算法。

**（2）分析**（必须手写！！！）

**（3）实验代码**（源代码可以打印，算法描述必须手写！！！）

#include<bits/stdc++.h>

using namespace std;

const int board=100;

int main(){

long long map[board][board];

int dis[board],book[board],u,v,t1,t2,t3,x,y,min,s;

vector<int> path[board];

long long inf=99999;

for(int i=0;i<board;i++){

for(int j=0;j<board;j++){

map[i][j]=inf;

if(i==j){

map[i][j]=0;

}

}

}

cin>>u>>v>>s;

for(int i=1;i<=v;i++){

cin>>t1>>t2>>t3;

map[t1][t2]=t3;

}

for(int i=1;i<=u;i++){

dis[i]=map[s][i];

}

book[1]=1;

for(int i=1;i<u;i++){

min=inf;

for(int j=1;j<=u;j++){

if(book[j]==0&&dis[j]<min){

min=dis[j];

x=j;

}

}

book[x]=1;

for(y=1;y<=u;y++){

if(map[x][y]<inf){

if(dis[y]>dis[x]+map[x][y]){

dis[y]=dis[x]+map[x][y];

path[y].push\_back(x);

}

}

}

}

for(int i=1;i<=u;i++){

cout<<s<<"to"<<i<<"length:"<<dis[i]<<endl;

cout<<"path is"<<i<<",";

for(int j=0;j<path[i].size();j++){

cout<<path[i][j]<<",";

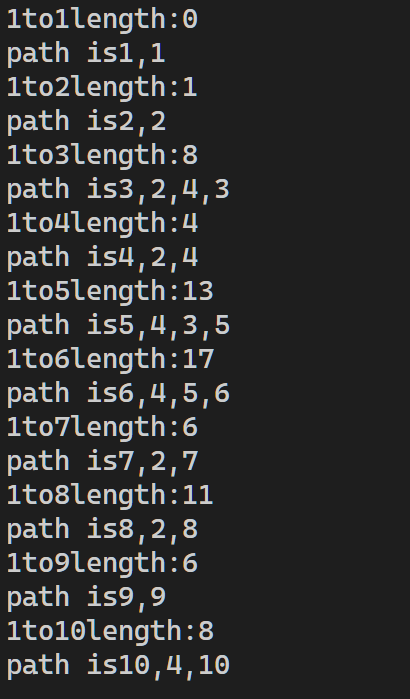
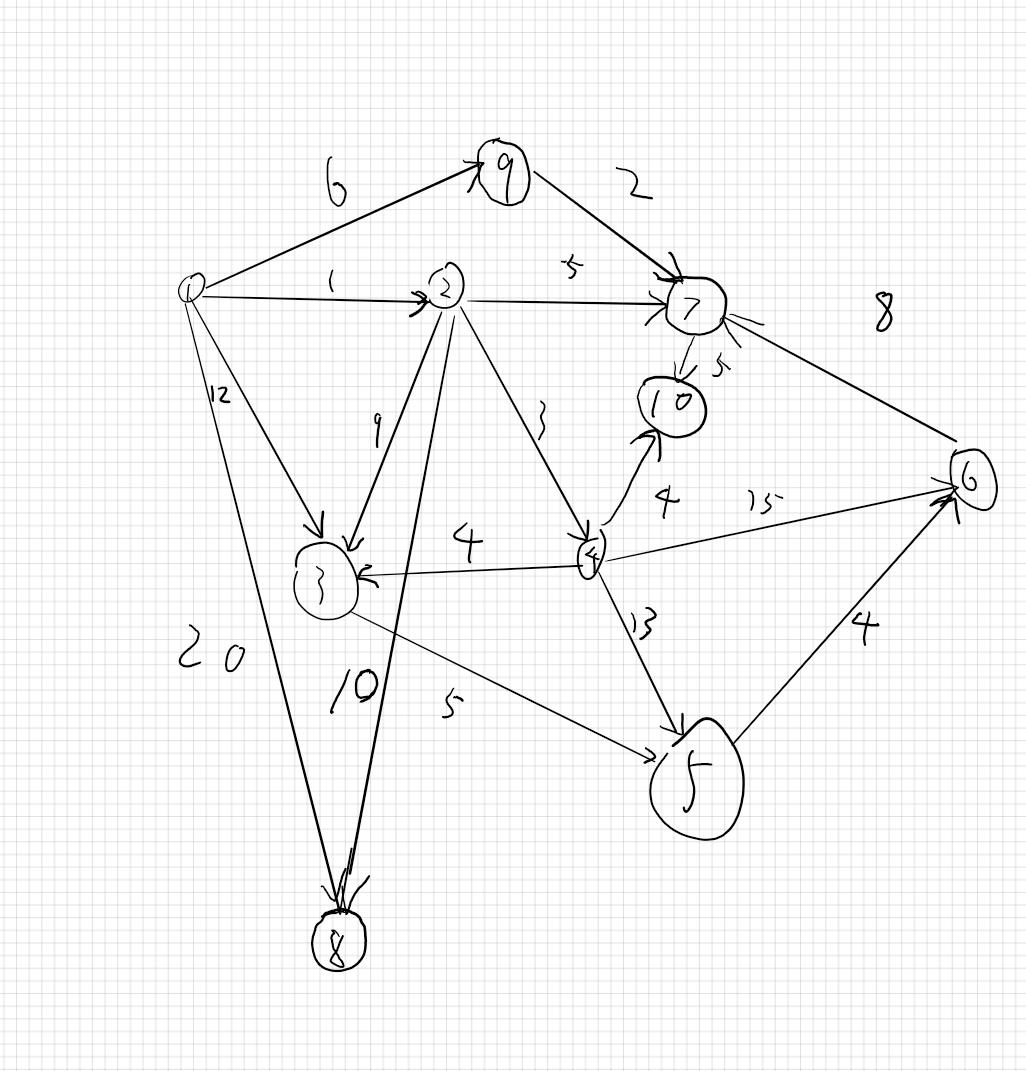
}

cout<<i<<endl;

}

return 0;

}



**三、实验小结**（必须手写！！！）

实验小结部分必须手写，主要是总结编程中出现的问题，解决的方法，有关数据结构的思考。