**暨南大学本科实验报告专用纸**

课程名称 数据结构实验 成绩评定

实验项目名称 求最短路径 指导教师 刘波、郭科芩

实验项目编号 8060154909 实验项目类型 综合性 实验地点 N116

学生姓名 阮炜霖 学号 2020101603

学院 信息科学技术学院 系 计算机系 专业 网络工程

实验时间 2021 年 10 月 9 日 上 午～ 10 月 9 日 上 午

**（一）实验目的和要求；**

目的：掌握图的基本存储方法，了解最短路径的概念，掌握求最短路径的方法

要求：任选一种高级程序语言编写源程序，并调试通过，测试正确。

**（二）实验主要内容；**

（1）建立一个包含6个结点的有向图，

（2）利用Dijkstra算法求顶点v0到其它顶点的最短路径。

**（三）实验原理**

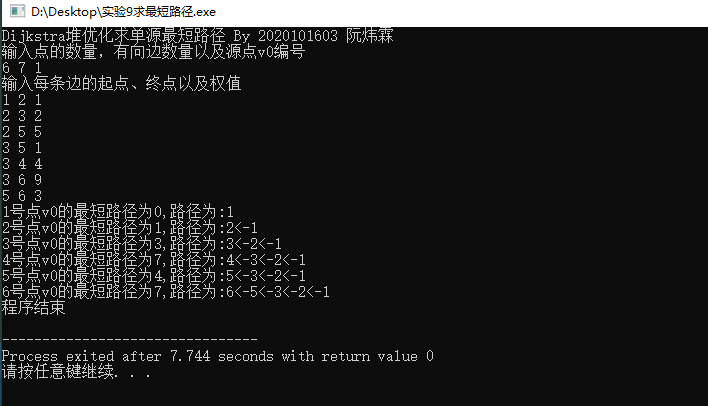
利用链式前向星建立有向图，然后从源点出发，用Dijkstra算法求单源最短路，并记录每个节点的前一个所经点以求出路径。再加上堆优化将复杂度优化成O(nlogn)。

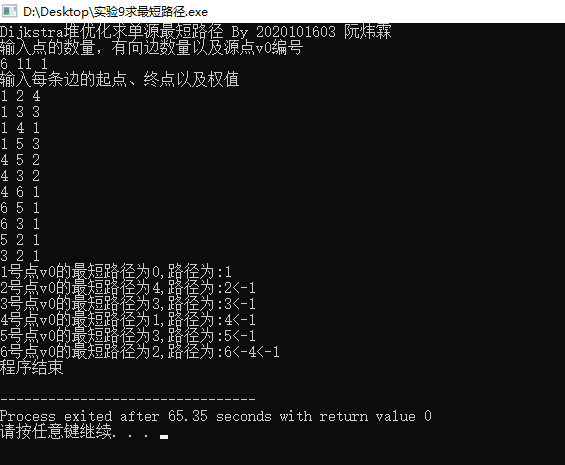
**（四）实验步骤及调试分析；**

通过链式前向星定义有向边的存储方式，随后编写Dijkstra算法实现求单源最短路及其路径，之后对程序进行不断调试得出正解。查阅资料得知Dijkstra可通过堆优化的方式将时间复杂度由O(n^2)优化到近似O(nlogn)，对程序不断修改最终完成优化。

**（五）实验结果及分析；**

通过实验结果，我们得出结论：该程序使用Dijkstra堆优化算法实现了求最短路径。





**（六）附录：源程序**

#include<bits/stdc++.h>

#define M(x,y) make\_pair(x,y)

using namespace std;

const int N=100005;

int tot=0,head[N],vis[N],pre[N];

int n,m,s,u,v,w;

long long dir[N];

struct Edge{

int to,next,dis;

}edge[N<<1];

void addedge(int u,int v,int w){

edge[++tot].next=head[u];

edge[tot].to=v;

edge[tot].dis=w;

head[u]=tot;

}

struct node{

int dis;int pos;

bool operator <(const node &x)const{

return x.dis<dis;

}

};

void dij(int s){

memset(vis,0,sizeof(vis));

memset(dir,127/3,sizeof(dir));

dir[s]=0;

vis[s]=0;

priority\_queue<node>q;

q.push(node{0,s});

while(!q.empty()){

int x=q.top().pos;

q.pop();

if(vis[x]) continue;

vis[x]=1;

for(int i=head[x];i;i=edge[i].next){

int to=edge[i].to;

if(dir[to]>dir[x]+edge[i].dis){

dir[to]=dir[x]+edge[i].dis;

pre[to]=x;

if(!vis[to]){

q.push(node{dir[to],to});

}

}

}

}

}

void print\_path(int ed){

int j=ed;

while(j!=s){

printf("%d<-",j);

j=pre[j];

}

printf("%d\n",j);

}

int main(){

printf("Dijkstra堆优化求单源最短路径 By 2020101603 阮炜霖\n");

printf("输入点的数量，有向边数量以及源点v0编号\n");

scanf("%d%d%d",&n,&m,&s);

printf("输入每条边的起点、终点以及权值\n");

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

scanf("%d%d%d",&u,&v,&w);

addedge(u,v,w);

}

dij(s);

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

printf("%d号点v0的最短路径为%d,路径为:",i,dir[i]);

print\_path(i);

}

printf("程序结束\n");

return 0;

}

**暨南大学本科实验报告专用纸(附页)**