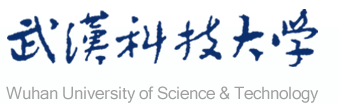
****

《数据结构》实验报告

**专 业：**

**班 级：**

**学 号：**

**姓 名：**

**一、【实验标题】**

**实验4 图（公交线路图）**

**二、【实验目的】**

1、掌握图的定义和图的邻接表存储结构。

2、掌握图的创建方法。

3、掌握顶点和边的操作。

4、掌握图的基本算法并能实际应用。

5、掌握图的深度优先搜索算法以及实现方法。

6、使用C/C++语言和图实现“公交线路图”专题。

**三、【实验环境】**

硬件: Intel®Core™i5-6500 CPU3.20GHz,8GB内存

软件: Windows 7 64位,Microsoft Visual Studio 2010

**四、【实验内容】**

本程序是一个控制台程序，模拟城市公交系统，程序中保存了城市的公交线路和公交站点信息。公交线路和站点满足一下条件：

1. 某些站点之间有一条或者多条公交线路直接到达，即联通。
2. 每条公交线路经过若干站点。
3. 相同的行径路线都有上行和下行两条公交路线，公交路线都是单向行驶的（有向图）。

**五、【实验程序】**

**/\***

本程序使用邻接表建立无向的公交线路图，利用了dfs和dijkstra 算法实现查询路径和输出最短路径。

**\*/**

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int N=1e3+5,M=2e4+5,inf=0x3f3f3f3f,mod=1e9+7;

#define mst(a,b) memset(a,b,sizeof a)

#define PII pair<int,int>

#define fi first

#define se second

#define pb push\_back

// string name[N];

// map<string,int>mp;

int cnt,h[N],st,ed,vis[N],in[N];

int tot;

vector<int>path;

struct edge{

int to,nt,w;

}e[N<<1];

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

e[++cnt]={v,h[u],w},h[u]=cnt;

e[++cnt]={u,h[v],w},h[v]=cnt;

}

void dfs(int u,int fa){

// printf("u=%d,fa=%d\n",u,fa);

if(u==ed){

tot++;

printf("第 %d 条路径为:\n",tot);

printf("%d",st);

for(int v:path) printf(" -> %d",v);

printf("\n");

return;

}

for(int i=h[u];i;i=e[i].nt){

int v=e[i].to;

if(v==fa||vis[v]) continue;

path.pb(v);

vis[v]=1;

dfs(v,u);

vis[v]=0;

path.pop\_back();

}

}

int d[N],pre[N];

stack<int>sk;

void dij(int st,int ed){

queue<int>q;

mst(d,0x3f),d[st]=0;

q.push(st);in[st]=1;

while(!q.empty()){

int u=q.front();q.pop();in[u]=0;

for(int i=h[u];i;i=e[i].nt){

int v=e[i].to,w=e[i].w;

if(d[v]>d[u]+w){

d[v]=d[u]+w;

pre[v]=u;

if(!in[v]) q.push(v),in[v]=1;

}

}

}

while(ed!=st){

sk.push(ed);

ed=pre[ed];

}

}

int main(){

int n,m;

printf("请建立公交线路图\n");

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

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

int u,v,w;

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

add(u,v,w);

}

printf("\n已经使用邻接表完成公交线路图的构建\n\n");

printf("请输入查询的起点和终点:\n");

scanf("%d%d",&st,&ed);

vis[st]=0,dfs(st,0);

dij(st,ed);

printf("\n从 %d 到 %d 的路径个数为: %d\n\n",st,ed,tot);

printf("从起点 %d 到 终点 %d 的最短路线是:\n",st,ed);

printf("%d",st);

while(!sk.empty()){

printf(" -> %d",sk.top());

sk.pop();

}printf("\n\n");

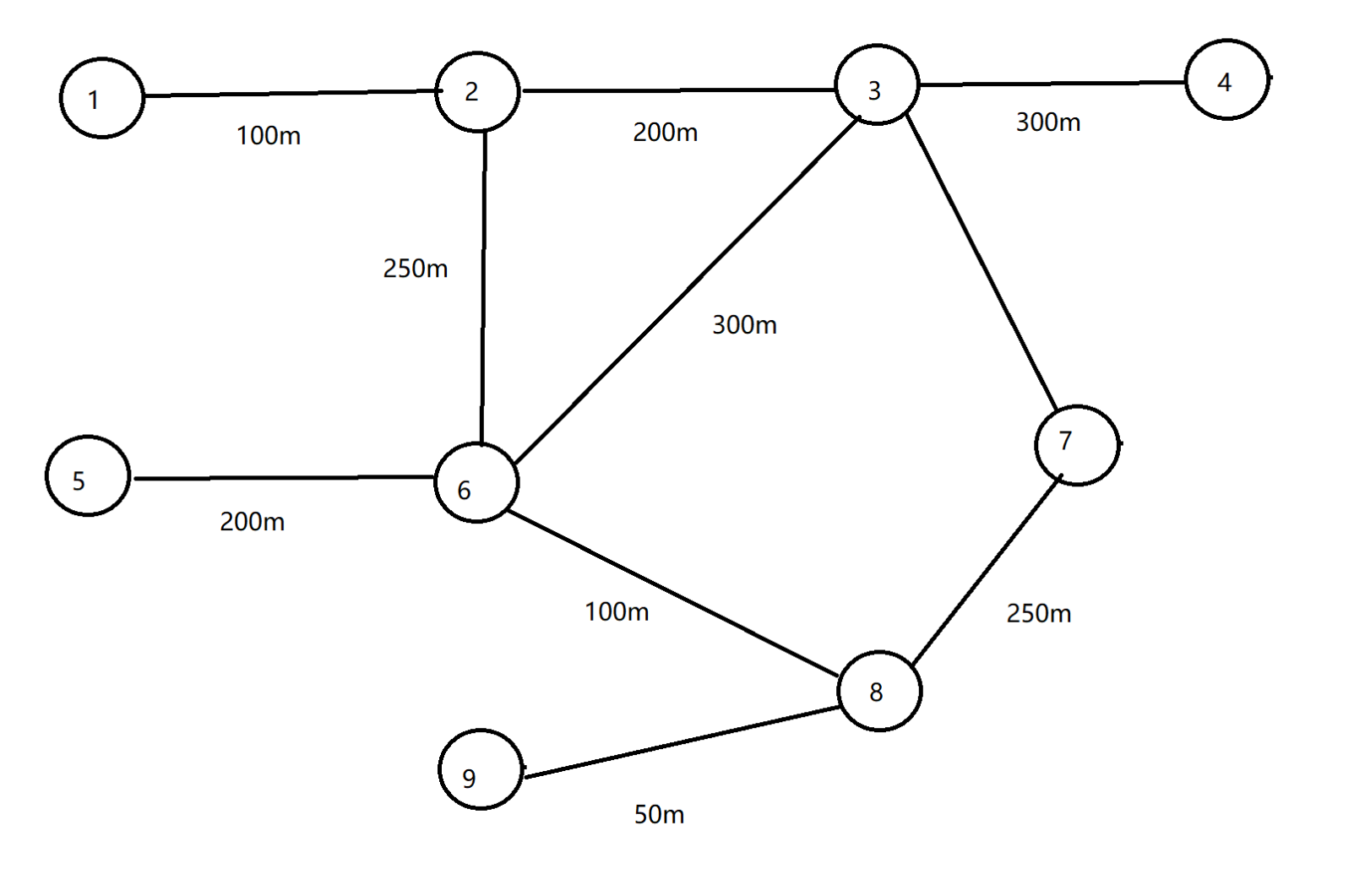
printf("该该线路的长度为: %d\n",d[ed]);

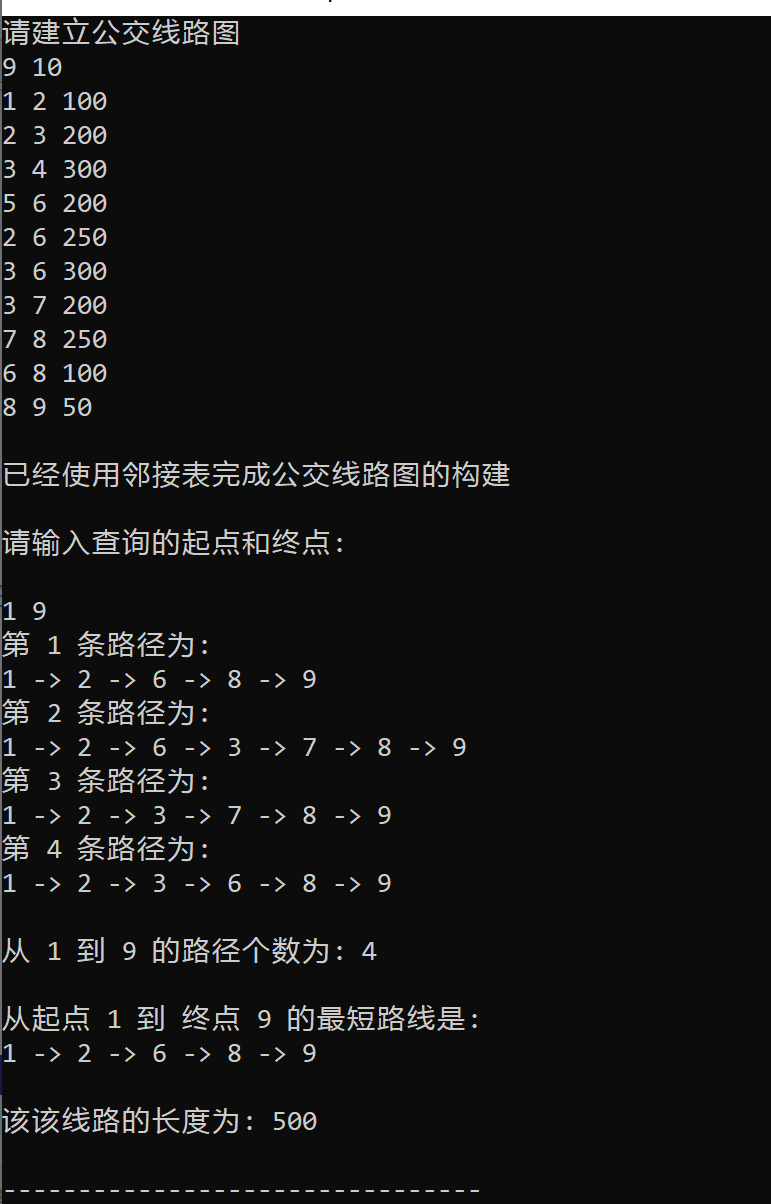
return 0;

}

**六、【实验运行结果】**

**公交线路图**





**七、【实验小结】**

**本次实验我学会了关于图的定义和图的邻接表存储结构的知识，明白了图所包含的几个要素，知道了点和边的定义，并且掌握了多种图的创建方法，如邻接矩阵建图，邻接表建图，链式前向星建图。知道了如果对图的顶点和边进行操作，学会了使用dfs,bfs算法来遍历整个图，通过队列或者数组来保存结点的路径，能够熟练地运用dijkstra或者spfa算法计算图的最短路径问题，通过维护数组在dfs的过程中输出多种路径。**

**通过本次实验的学习，我对于图论的知识有了更深刻的理解，也让我明白了数据结构中的图论可以与现实生活中的种种问题相结合，使我对于算法融入到生活实际中有了更熟练的掌握能力。**