两景点最短路径模块

### 代码

typedef int path\_d[M];

typedef int dist\_d[M];

typedef int path\_f[M][M];

typedef int dist\_f[M][M];

typedef int st\_d[M];

typedef int st\_f[M];

void floyd(Mgraph g,path\_f p,dist\_f d)

{

int i,j,k;

for(i=0;i<g.n;i++)

for(j=0;j<g.n;j++)

{

d[i][j]=g.edges[i][j];

if(i!=j&&d[i][j]<FINITY)

p[i][j]=i;

else p[i][j]=-1;

}

for(k=0;k<g.n;k++)

{

for(i=0;i<g.n;i++)

for(j=0;j<g.n;j++)

if(d[i][j]>d[i][k]+d[k][j])

{

d[i][j]=d[i][k]+d[k][j];

p[i][j]=k;

}

}

}

void dijkstra(Mgraph g,int v0,path\_d p,dist\_d d)

{

boolean final[M];

int i,k,v,min;

for(v=0;v<g.n;v++)

{

final[v]=FALSE;

d[v]=g.edges[v0][v];

if(d[v]<FINITY&&d[v]!=0)

p[v]=v0;else p[v]=-1;

}

final[v0]=TRUE;d[v0]=0;

for(i=1;i<g.n;i++)

{

min=FINITY;

for(k=0;k<g.n;++k)

if(!final[k]&&d[k]<min)

{

v=k;min=d[k];

}

if(min==FINITY) return;

final[v]=TRUE;

for(k=0;k<g.n;++k)

if(!final[k]&&(min+g.edges[v][k]<d[k]))

{

d[k]=min+g.edges[v][k];

p[k]=v;

}

}

}

void print\_gpd(Mgraph g,int fn,path\_d p,dist\_d d,st\_d s)

{

int pre,top=-1;

s[++top]=fn;

pre=p[fn];

while(pre!=-1)

{

s[++top]=pre;

pre=p[pre];

}

}

void print\_floyd(Mgraph g,int bgn,int fnl,path\_f p,dist\_f d, st\_f s)

{

int pre,top=-1;

s[++top]=fnl;

pre=p[bgn][fnl];

while(pre!=-1)

{

s[++top]=pre;

pre=p[bgn][pre];

}

}