



导航

- [首页](#)
- [社区主页](#)
- [当前事件](#)
- [最近更新](#)
- [随机页面](#)
- [使用帮助](#)
- [NOCOW地图](#)
- [新手试练场](#)

搜索

工具箱

- [链入页面](#)
- [链出更改](#)
- [特殊页面](#)
- [可打印版](#)
- [永久链接](#)

为防止广告，目前nocow只有登录用户能够创建新页面。如要创建页面请先[登录/注册](#)（新用户需要等待1个小时才能正常使用该功能）。

朱永津刘振宏算法

朱永津刘振宏算法是一个确定有向图最小树形图的算法

对于一个有向图 $G=(V,E)$ ，首先需要确定最小树形图的根。对于未确定根的图，我们枚举所有点为根并找最优的解。

下面假设根已确定，那么寻找最小树形图的算法为：

- $ans=0$;
- 对于除了根节点外的每个节点 i ，求出权值最小的入边 $(prev[i], i)$ 。如果这些入边不构成环，那么入边的集合 E 即为所求
- 否则,假设其中的一个环为 $v_0 \leftarrow v_1 \leftarrow \dots \leftarrow v_k \leftarrow v_0$ ，那么更新原图: $w(v[0], u) = \min\{w(v[j], u)\} (0 \leq j \leq k)$

$w(u, v[0]) = \min\{w(u, v[j]) - w(prev[j], j)\} (0 \leq j \leq k)$ 并且 ans 的值增加环中所有边的权值

- 重复以上操作，直到 E 中不存在环。此时 ans 就是最小树形图的边权值之和。

如果采用邻接表存储，那么以上算法的时间复杂度为 $O(VE)$ 。如果采用邻接矩阵存储，那么以上算法的时间复杂度为 $O(V^3)$ 。

因此，整个最小树形图的算法可以在 $O(V^2 \cdot E)$ 的时间内完成

例题:POJ3164

```
/*
problem: poj3164
author: gnagnoyil
*/
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <cmath>
#define maxn 103
#define maxlongint (1<<30)
double g[maxn][maxn];
int prev[maxn],x[maxn],y[maxn];
bool flag[maxn],ha[maxn];
int n,m;
double ans=0;
inline double sqr(double a){
    return a*a;
}
double dis(int x1,int y1,int x2,int y2){
    return sqrt(sqr(x2-x1)+sqr(y2-y1));
}
int dfs(int x){
    ha[x]=true;
    for(int i=1;i<=n;i++){
        if((!ha[i])&&(g[x][i]<maxlongint))
            dfs(i);
    }
    return 0;
}
int main(){
    for(;scanf("%d%d",&n,&m)!=EOF;){
        memset(prev,0,sizeof(prev));
        memset(flag,false,sizeof(flag));
        memset(ha,false,sizeof(ha));
        memset(x,0,sizeof(x));
        memset(y,0,sizeof(y));
        ans=0;
        for(int i=1;i<=n;i++)
            for(int j=1;j<=n;j++)
                g[i][j]=maxlongint;
        for(int i=1;i<=n;i++)
            scanf("%d%d",&x[i],&y[i]);
        for(int a,b;m>0;m--){
            scanf("%d%d",&a,&b);
```

```

    g[a][b]=dis(x[a],y[a],x[b],y[b]);
}
dfs(1);
bool bb=false;
for(int i=1;i<=n;i++)
    if(!ha[i]){
        printf("poor snoopy\n"),bb=true;
        break;
    }
if(bb)
    continue;
for(,true;){
    int i;
    for(i=2;i<=n;i++)
        if(!flag[i]){
            prev[i]=i,g[i][i]=maxlongint;
            for(int j=1;j<=n;j++)
                if((!flag[j])&&(g[j][i]<g[prev[i]][i]))
                    prev[i]=j;
        }
    for(i=2;i<=n;i++)
        if(!flag[i]){
            int j=i;
            memset(ha,false,sizeof(ha));
            ha[1]=true;
            for(,!ha[prev[j]];ha[j]=true,j=prev[j]);
            j=prev[j];
            if(j==1)
                continue;
            ans+=g[prev[j]][j],i=j;
            for(j=prev[i];j!=i;j=prev[j])
                flag[j]=true,ans+=g[prev[j]][j];
            for(j=1;j<=n;j++)
                if((!flag[j])&&(g[j][i]<maxlongint))
                    g[j][i]=g[prev[i]][i];
            for(int k=1;k<=n;k++)
                if(!flag[k])
                    for(j=prev[i];j!=i;j=prev[j]){
                        if(g[j][k]<g[i][k])
                            g[i][k]=g[j][k];
                        if((g[k][j]<maxlongint)&&(g[k][j]-g[prev[j]][j]<g[k][i]))
                            g[k][i]=g[k][j]-g[prev[j]][j];
                    }
                break;
        }
    if(i>n){
        for(i=2;i<=n;i++)
            if(!flag[i])
                ans+=g[prev[i]][i];
        break;
    }
}
printf("%.2lf\n",ans);
}
return 0;
}
//Orz gnaggnoyil 神牛

```



此页面已被浏览过2,399次。 本页面由cosechy@gmail.com于2012年3月3日 (星期六) 05:00做出最后修改。
 在SuperBrother和NOCOW用户Wwwaaannnggrs的工作基础上。 本站全部文字内容使用GNU Free
 Documentation License 1.2授权。 隐私权政策 关于NOCOW 免责声明 陕ICP备09005692号

