

CCPC Wannafly Camp Day3

[比赛链接](#)

C. 无向图定向

题意

给定无向图，要求对每条边定向，得到 DAG 同时最小化最长路。

题解 1

易知可以当拓扑序确定时每条边方向是确定的，考虑枚举所有拓扑序的全排列然后计算最长路。

显然这样会 TLE 于是考虑随机化乱搞，居然过了。

```
const int MAXN=20,MAXM=200;
struct Edge{
    int to,next;
}edge[MAXM<<1];
int head[MAXN],edge_cnt;
void Insert(int u,int v){
    edge[++edge_cnt]=Edge{v,head[u]};
    head[u]=edge_cnt;
}
int deg[MAXN],dp[MAXN],a[MAXN];
vector<int> g[MAXN];
int cal(int n){
    int ans=0;
    queue<int> q;
    _rep(u,1,n){
        dp[u]=0;
        g[u].clear();
        for(int i=head[u];i;i=edge[i].next){
            int v=edge[i].to;
            if(a[u]>a[v]){
                g[u].push_back(v);
                deg[v]++;
            }
        }
    }
    _rep(i,1,n){
        if(!deg[i])
            q.push(i);
    }
}
```

```
while(!q.empty()){
    int u=q.front();
    q.pop();
    ans=max(ans,dp[u]);
    _for(i,0,g[u].size()){
        int v=g[u][i];
        dp[v]=max(dp[v],dp[u]+1);
        deg[v]--;
        if(!deg[v])
            q.push(v);
    }
}
return ans;
}
int main()
{
    int n=read_int();
    int m=read_int();
    _for(i,0,m){
        int u=read_int(),v=read_int();
        Insert(u,v);
        Insert(v,u);
    }
    int ans=n-1;
    _rep(i,1,n)a[i]=i;
    while((double)clock()/CLOCKS_PER_SEC<0.9){
        random_shuffle(a+1,a+n+1);
        ans=min(ans,cal(n));
    }
    enter(ans);
    return 0;
}
```

题解 2

考虑对无向图进行染色，使得同色点之间没有连边，最小化颜色种数。颜色代表数值高的点向颜色代表数值低的点连边，此时答案为染色数 -1 。

考虑 $O(n^2)$ 标记所有独立子集，然后 $O(3^n)$ 子集枚举计算染色数。

```
int G[20],dp[1<<20];
bool ok[1<<20];
int n,m;
int main()
{
    scanf("%d%d",&n,&m);
    ok[0]=true;
    while(m--){

```

```

    int x,y;
    scanf("%d%d",&x,&y);
    x--;y--;
    G[x]|=1<<y;G[y]|=1<<x;
}
for(int i=0;i<(1<<n);i++)
{
    for(int j=0;j<n;j++)
    {
        ok[i|(1<<j)] |= ok[i] && !(i&(1<<j)) && !(i&G[j]);
        //取出一个独立集
        //j这个点不在方案i中并且j和当前的集合没有直接相连的边
    }
}
for(int i=0;i<(1<<n);i++) dp[i]=1<<20;
dp[0]=0;
for(int i=1;i<(1<<n);i++)
{
    if(ok[i]) dp[i]=1; //独立集只需要一种颜色
    for(int j=i&(i-1);j=i&(j-1))
    {
        if(ok[j])
            dp[i]=min(dp[i],dp[i^j]+1); //取出一个独立集给一种新颜色
        if(j==0) break;
    }
}
printf("%d\n",dp[(1<<n)-1]-1);
return 0;
}

```

G. 火山哥周游世界

题意

给定一棵边权树和 k 个关键点，问从第 $i(1 \leq i \leq n)$ 点出发经过所有关键点的最短路程。

题解

不难发现答案为第 i 个点与所有关键点构成的生成树的边权之和的两倍 - 第 i 个点到最远关键点的距离。

换根 dp 维护每个结点的生成树边权和，每个结点的子树方向的最远关键结点、次远关键结点以及根节点方向的最远关键结点即可。

时间复杂度 $O(n)$

```
const int MAXN=5e5+5;
```

```
LL inf=1e15;
struct Edge{
    int to,w,next;
}edge[MAXN<<1];
int head[MAXN],edge_cnt,k;
void Insert(int u,int v,int w){
    edge[++edge_cnt]=Edge{v,w,head[u]};
    head[u]=edge_cnt;
}
LL f[MAXN],dp[MAXN][3],ans[MAXN];
int sz[MAXN],hson[MAXN];
void dfs1(int u,int fa){
    hson[u]=0;
    for(int i=head[u];i;i=edge[i].next){
        int v=edge[i].to;
        if(v==fa)continue;
        dfs1(v,u);
        sz[u]+=sz[v];
        if(sz[v])f[u]+=f[v]+edge[i].w*2;
        if(dp[v][0]+edge[i].w>dp[u][0]){
            hson[u]=v;
            dp[u][1]=dp[u][0];
            dp[u][0]=dp[v][0]+edge[i].w;
        }
        else if(dp[v][0]+edge[i].w>dp[u][1])
            dp[u][1]=dp[v][0]+edge[i].w;
    }
}
void dfs2(int u,int fa){
    ans[u]=f[u]-max(dp[u][0],dp[u][2]);
    for(int i=head[u];i;i=edge[i].next){
        int v=edge[i].to;
        if(v==fa)continue;
        if(sz[v]==0)
            f[v]=f[u]+edge[i].w*2;
        else if(sz[v]!=k)
            f[v]=f[u];
        if(v==hson[u])
            dp[v][2]=edge[i].w+max(dp[u][1],dp[u][2]);
        else
            dp[v][2]=edge[i].w+max(dp[u][0],dp[u][2]);
        dfs2(v,u);
    }
}
int main()
{
    int n=read_int();
    k=read_int();
    _for(i,1,n){
        int u=read_int(),v=read_int(),w=read_int();
    }
}
```

```
    Insert(u,v,w);
    Insert(v,u,w);
}
_rep(i,1,n)
dp[i][0]=dp[i][1]=dp[i][2]=-inf;
_for(i,0,k){
    int u=read_int();
    dp[u][0]=dp[u][1]=dp[u][2]=0;
    sz[u]=1;
}
dfs1(1,0);
dfs2(1,0);
_rep(i,1,n)
enter(ans[i]);
return 0;
}
```

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:ccpc_wannafly_winter_camp_day3&rev=1619875516

Last update: 2021/05/01 21:25