

Educational Codeforces Round 96

[比赛链接](#)

G. Yet Another DAG Problem

题意

给定一个有向无环带边权图，要求给每个点赋值 a_i 对边 $u \rightarrow v$ 要求 $a_u \geq a_v$ 且最小化 $\sum_{i=1}^m w_i(a_{u_i} - a_{v_i})$

题解

首先，假设 a_i 的取值不连续，不妨设不存在 $a_u = a_v$ 于是将 $a_u > a_v$ 的点权值减一，则答案仍然合法，且 $\sum_{i=1}^m w_i(a_{u_i} - a_{v_i})$ 不减。

于是不妨假设 a_i 的取值必定为连续的一段，于是 $a_i \in [0, n-1]$

$$\sum_{i=1}^m w_i(a_{u_i} - a_{v_i}) = \sum_{i=1}^n \left(\sum_{u_j=i} w_j - \sum_{v_j=i} w_j \right) a_i = \sum_{i=1}^n c_i a_i$$

于是只需要考虑每个点的最优点权分配即可，考虑构建最小割模型。

首先令 $e_{i,j}$ 表示 $a_i = j$ 于是令该边的容量即删除该边的代价为 c_i 记 $b_{i,j} = u_{e_{i,j}}$

考虑连边 $s \rightarrow b_{i,0} \rightarrow b_{i,1} \rightarrow \dots \rightarrow b_{i,n-1} \rightarrow t$ 其中 $s \rightarrow b_{i,0}, b_{i,n-1} \rightarrow t$ 边权为 ∞ 该链上必须选择一条边。

接下来考虑维护 $a_u \geq a_v$ 的约束，于是连边 $b_{v,i} \rightarrow b_{u,i+1}$ 并将其容量设为 ∞

表示如果选择 $e_{v,i}$ 则考虑链 $s \rightarrow b_{v,0} \rightarrow b_{v,1} \rightarrow \dots \rightarrow b_{v,i} \rightarrow b_{u,i+1} \rightarrow \dots \rightarrow t$

该链上必须选择一条边，而选择了 $e_{v,i}$ 则不需要考虑 $e_{v,j} (j \leq i)$ 于是 $e_{u,j} (j \geq i)$ 必须选一条。

最后由于部分点 $c_i \leq 0$ 考虑为图中每条边加上一个偏移量。

由于最小割至少要包含 n 条边，且该模型必须只选择 n 条边，于是不影响正确性。

图中有 $O(n^2)$ 个点 $O(nm)$ 条边，于是时间复杂度为 $O(n^5m)$

```
const int MAXS=20,MAXN=405,MAXM=5005,base=1e7;
const LL Inf=1LL<<62;
struct Edge{
    int to,next;
    LL cap;
    Edge(int to=0,LL cap=0,int next=0){
```

```
        this->to=to;
        this->cap=cap;
        this->next=next;
    }
}edge[MAXM<<1];
int head[MAXN],edge_cnt;
void Clear(){mem(head,-1);edge_cnt=0;}
void Insert(int u,int v,LL c){
    edge[edge_cnt]=Edge(v,c,head[u]);
    head[u]=edge_cnt++;
    edge[edge_cnt]=Edge(u,0,head[v]);
    head[v]=edge_cnt++;
}
struct Dinic{
    int s,t;
    int pos[MAXN],vis[MAXN],dis[MAXN];
    bool bfs(int k){
        queue<int>q;
        q.push(s);
        vis[s]=k,dis[s]=0,pos[s]=head[s];
        while(!q.empty()){
            int u=q.front();q.pop();
            for(int i=head[u];~i;i=edge[i].next){
                int v=edge[i].to;
                if(vis[v]!=k&&edge[i].cap){
                    vis[v]=k,dis[v]=dis[u]+1,pos[v]=head[v];
                    q.push(v);
                    if(v==t)
                        return true;
                }
            }
        }
        return false;
    }
    LL dfs(int u,LL max_flow){
        if(u==t||!max_flow)
            return max_flow;
        LL flow=0,temp_flow;
        for(int &i=pos[u];~i;i=edge[i].next){
            int v=edge[i].to;
            if(dis[u]+1==dis[v]&&(temp_flow=dfs(v,min(max_flow,edge[i].cap)))){
                edge[i].cap-=temp_flow;
                edge[i^1].cap+=temp_flow;
                flow+=temp_flow;
                max_flow-=temp_flow;
                if(!max_flow)
                    break;
            }
        }
        return flow;
    }
}
```

```

    }
    int Maxflow(int s,int t){
        this->s=s;this->t=t;
        LL ans=0;
        int k=0;
        mem(vis,0);
        while(bfs(++k))
            ans+=dfs(s,Inf);
        return k;
    }
} solver;
int s[MAXS],idx[MAXS][MAXS],ans[MAXS];
int main()
{
    int n=read_int(),m=read_int(),be=1,en=2,node_cnt=3;
    _rep(i,1,n)_for(j,0,n)idx[i][j]=node_cnt++;
    Clear();
    while(m--){
        int u=read_int(),v=read_int(),w=read_int();
        s[u]+=w,s[v]-=w;
        _for(i,1,n)
            Insert(idx[v][i-1],idx[u][i],Inf);
    }
    _rep(i,1,n){
        Insert(be,idx[i][0],Inf);Insert(idx[i][n-1],en,Inf);
        _for(j,1,n)
            Insert(idx[i][j-1],idx[i][j],base+s[i]*(j-1));
    }
    int k=solver.Maxflow(be,en);
    _rep(i,1,n){
        _for(j,0,n){
            if(solver.vis[idx[i][j]]==k)
                ans[i]=j;
        }
    }
    _rep(i,1,n)
        space(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:edu_96

Last update: 2020/10/16 09:39