

[比赛链接](#)

补题情况

题目	蒋贤蒙	王赵安	王智彪
B	2	0	0
D	0	0	0
E	2	0	0
G	0	0	0
I	0	0	0
J	2	0	0
L	2	0	0

题解

J. Jazz Journey

题意

给定固定路线 $a_1, a_2 \dots a_d (1 \leq a_i \leq n)$ 需要坐飞机依次经过各点。

接下来给定若 m 种机票，每张机票有一个起点 u 和终点 v 以及费用 w

机票分为单程票和双程票，其中每张单程票只能实现一次 $u \rightarrow v$ 双程票可以实现一次 $u \rightarrow v$ 和一次 $v \rightarrow u$

题解

由于只有 $d-1$ 次移动，所以不妨将所有移动按起点终点分类，其中 $u \rightarrow v, v \rightarrow u$ 视为同类。

例如，对路径 12313121 ，可以得到如下几类：

- $1 \rightarrow 2, 1 \rightarrow 2, 2 \rightarrow 1$
- $2 \rightarrow 3$
- $3 \rightarrow 1, 1 \rightarrow 3, 3 \rightarrow 1$

然后对每类路径，单独考虑费用。不妨设某类路径的起点终点为 u, v 且 $u \rightarrow v$ 的双程票是最划算的。

考虑扫描该类型的所有移动，一旦出现 $u \rightarrow v, v \rightarrow u$ 则立刻购买 $u \rightarrow v$ 双程票，否则先保留。

最后剩余的移动一定是形如 $v \rightarrow u, v \rightarrow u \dots v \rightarrow u, u \rightarrow v \dots u \rightarrow v$ 这个时候再考虑买 $v \rightarrow u$ 的双程票划算还是单程票划算即可。

最后不要忘了把双程票当单程票用比单程票便宜的情况。时间复杂度 $O(d \log d)$

```
const LL inf=2e9;
```

```
const int MAXN=3e5+5;
map<pair<int,int>,int>mp;
struct Node{
    int u,v;
    int type;
    bool operator < (const Node &b)const{
        if(u!=b.u)
            return u<b.u;
        else if(v!=b.v)
            return v<b.v;
        else
            return type<b.type;
    }
};
map<Node,LL> cost;
vector<int> c[MAXN];
int a[MAXN];
LL ans;
void solve(int id,int u,int v){
    LL w1,w2,w3,w4;
    int cnt[2]={0,0};
    w1=cost.count(Node{u,v,0})?cost[Node{u,v,0}]:inf;
    w2=cost.count(Node{u,v,1})?cost[Node{u,v,1}]:inf;
    w3=cost.count(Node{v,u,0})?cost[Node{v,u,0}]:inf;
    w4=cost.count(Node{v,u,1})?cost[Node{v,u,1}]:inf;
    w1=min(w1,w2);
    w3=min(w3,w4);
    for(int t:c[id]){
        if(t==0){
            if(cnt[1]){
                if(w4==min(w2,w4)&&w4<w3+w1){
                    ans+=w4;
                    cnt[1]--;
                }
            }
            else
                cnt[0]++;
        }
        else
            cnt[0]++;
    }
    else{
        if(cnt[0]){
            if(w2==min(w2,w4)&&w2<w1+w3){
                ans+=w2;
                cnt[0]--;
            }
            else
                cnt[1]++;
        }
        else
    }
```

```

        cnt[1]++;
    }
}
int tt=min(cnt[0],cnt[1]);
if(w2!=min(w2,w4)&&w2<w1+w3){
    ans+=1LL*tt*w2;
    cnt[0]-=tt;
    cnt[1]-=tt;
}
else if(w4!=min(w2,w4)&&w4<w3+w1){
    ans+=1LL*tt*w4;
    cnt[0]-=tt;
    cnt[1]-=tt;
}
ans+=1LL*cnt[0]*w1+1LL*cnt[1]*w3;
}
int main(){
    int n=read_int(),d=read_int();
    _for(i,0,d)a[i]=read_int();
    _for(i,1,d){
        int x=a[i-1],y=a[i];
        if(x>y)swap(x,y);
        if(mp.find(make_pair(x,y))==mp.end()){
            int t=mp.size();
            mp[make_pair(x,y)]=t;
        }
        c[mp[make_pair(x,y)]] .push_back(a[i-1]>a[i]);
    }
    int m=read_int();
    _for(i,0,m){
        int u=read_int(),v=read_int();
        char type=get_char();
        LL w=read_int();
        Node t;
        if(type=='0')
            t=Node{u,v,0};
        else
            t=Node{u,v,1};
        if(cost.find(t)==cost.end())
            cost[t]=inf;
        cost[t]=min(cost[t],w);
    }
    for(map<pair<int,int>,int>::iterator
iter=mp.begin();iter!=mp.end();iter++){
        pair<int,int> temp=iter->first;
        solve(iter->second,temp.first,temp.second);
    }
    enter(ans);
    return 0;
}

```

L. Lost Logic

题意

给定 n 个变量以及三组解，要求构造若干条形如 $(!x_i \text{ to } !x_j)$ 的限制，使得方程只有给定的三组解。

题解

将所有变量在三组解中的取值分为 8 类，分别是 $(0,0,0), (0,0,1) \cdots (1,1,1)$

首先对于 $(0,0,0)$ 类变量，构造约束 $x \text{ to } !x$ 类似处理 $(1,1,1)$ 类变量。

对于同属于 $(0,0,1)$ 类变量 x, y 构造约束 $x \text{ to } y, !x \text{ to } !y$

于是现在自由变量最多只有 6 个。然后处理对偶的自由变量，如 $x \in (0,0,1), y \in (1,1,0)$ 构造约束 $x \text{ to } !y, !x \text{ to } y$

现在只剩下最多 3 个自由变量了。不难发现如果有 3 个自由变量则无法进一步构造约束。

当有两个自由变量时，不妨考虑 $x \in (0,0,1), y \in (1,0,0)$ 的情况，只要去除 $x=1, y=1$ 的情况即可，可以构造约束 $x \text{ to } !y$ 其他情况类似。

最后只剩下不超过一个自由变量，显然该变量任意取值后所有其他变量都取值固定，正好满足三组解。

```
const int MAXN=55;
int a[3][MAXN],b[2][2];
vector<int> c[2][2][2];
vector<pair<int,int> >ans;
int main(){
    int n=read_int();
    _for(i,0,3)_rep(j,1,n)
        a[i][j]=read_int();
    _rep(i,1,n)
        c[a[0][i]][a[1][i]][a[2][i]].push_back(i);
    vector<int> var;
    _for(i,0,2)_for(j,0,2)_for(k,0,2){
        if(c[i][j][k].size()==0)continue;
        if(i==j&& i==k){
            for(int t:c[i][j][k]){
                if(i==0)
                    ans.push_back(make_pair(t,t+n));
                else
                    ans.push_back(make_pair(t+n,t));
            }
        }
        else{
            int head=*c[i][j][k].begin();
            for(int t:c[i][j][k]){
                if(t==head)continue;
            }
        }
    }
}
```

```

        ans.push_back(make_pair(head,t));
        ans.push_back(make_pair(head+n,t+n));
    }
}
_for(i,0,2)_for(j,0,2){
    if(i==0&&j==0)continue;
    if(c[0][i][j].size()&&c[1][!i][!j].size()){
        int t1=*c[0][i][j].begin();
        int t2=*c[1][!i][!j].begin();
        ans.push_back(make_pair(t1,t2+n));
        ans.push_back(make_pair(t1+n,t2));
    }
    if(c[0][i][j].size())
        var.push_back(*c[0][i][j].begin());
    else if(c[1][!i][!j].size())
        var.push_back(*c[1][!i][!j].begin());
}
if(var.size()<3){
    if(var.size()==2){
        int t1=var[0],t2=var[1];
        int d1=2-a[0][t1]-a[1][t1]-a[2][t1],d2=2-a[0][t2]-a[1][t2]-
a[2][t2];
        ans.push_back(make_pair(t1+(1-d1)*n,t2+d2*n));
    }
    enter(ans.size());
    for(pair<int,int> t:ans){
        if(t.first>n){
            t.first-=n;
            putchar('!');
        }
        printf("x%d -> ",t.first);
        if(t.second>n){
            t.second-=n;
            putchar('!');
        }
        printf("x%d\n",t.second);
    }
}
else
puts("-1");
return 0;
}

```

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

Permanent link: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:%E7%BB%84%E9%98%9F%E8%AE%AD%E7%BB%83%E6%AF%94%E8%B5%9B%E8%AE%B0%E5%BD%95:contest11&rev=1628220556

Last update: 2021/08/06 11:29