

[比赛链接](#)

## 补题情况

| 题目 | 蒋贤蒙 | 王赵安 | 王智彪 |
|----|-----|-----|-----|
| B  | 0   | 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;
}

```

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=1628170693](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=1628170693)

Last update: 2021/08/05 21:38