芙蓉花

芙蓉王

放在前面：本题数据较随机，可能有优秀的贪心拿到比较高的分数。

Part 1

用搜索给每位顾客暴力分配糖果，可以拿到 $10pts$ 。

Part 2

不难建出网络流模型：S 向每种花连容量为 s_i 的边，对于每位常驻顾客，向 T 连 x_i 的边，并从区间 $[l_i,r_i]$ 的每种花向其连 inf 的边。
对于接下来的 Q 组询问，对于顾客 i 向 T 连 B_i 的边，并从每种花向 i 连 A_i 的边。
总边数为 $O(n(m+\sum k))$ 级别，可以通过 $25pts$ 。

Part 3

进入正片!
对于 Part 2 的网络流模型，我们考虑去模拟它。
经典的，有**最大流等于最小割**，于是可以将问题转变为求最小割。

Part 3.1

对于 $m=0$ 的数据，假设我们已经确定了左部点的割的情况，设割掉了 x 条边，先不考虑具体是哪 x 条，那么右部的每位顾客的贡献为 $min(a_i(n-x),b_i)$ ，也就是要么割掉连向 T 的 b_i ，要么割掉与左部 $(n-x)$ 个点相连的 $a_i(n-x)$ 。
于是发现，右部的最小代价只和左部割掉边的数量 x 有关，那么贪心地左边选最小的 x 条边割掉即可。右边的话，因为每个点的代价是个分段函数，且对于 x 的变化是单调的，按 $\frac{b_i}{a_i}$ 排序，再双指针即可。
复杂度 $O(nlogn+\sum(klogk)+Q(n+k))$ ，可以多拿 $10pts$

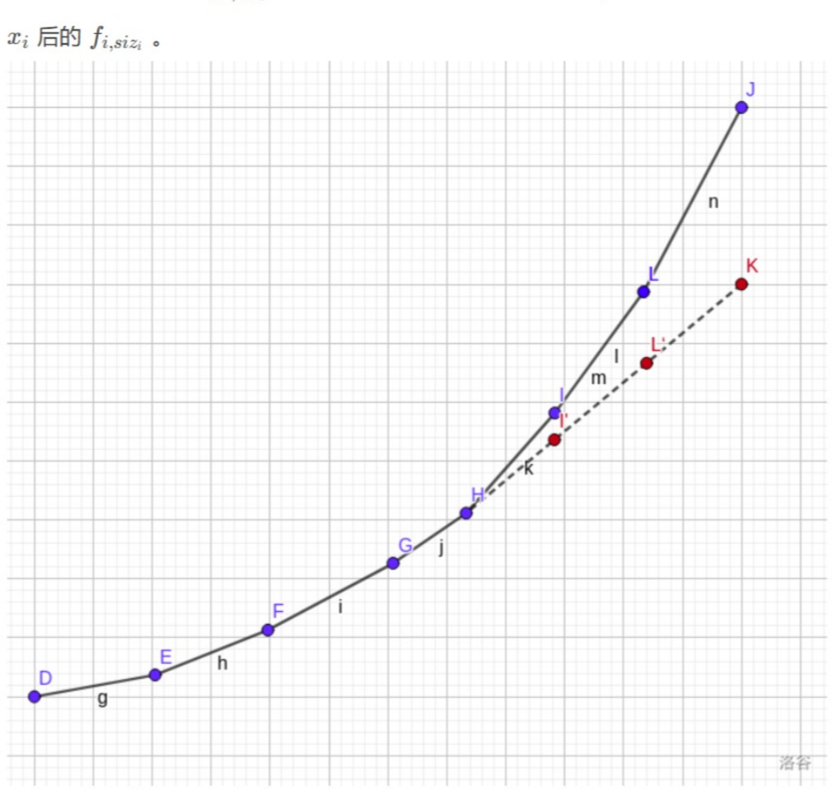
Part 3.2

考虑 $n,m,Q\le 2000$ 的数据，左边的若干个区间构成一棵树，每个树上的点有若干个可以割掉的左部点。
设 $f_{i,j}$ 表示在 i 的子树里割掉了 j 个左部点的最小代价。
特别的，如果一棵子树内的存在左部点没有被割掉，那么其代价会加上 x_i 。这样不好转移，于是转化一下，先把答案加上 $\sum x_i$ ，变成如果一棵子树内的左部点都被割掉，那么其代价会减去 x_i 。然后就有一个显然的DP。
求出 $f_{rt,1...k}$ 后，再像 Part 3.1的右边一样求答案即可。
时间复杂度 $O(n(m+Q)+\sum(klogk))$ ，可以把前 $40pts$ 拿了。

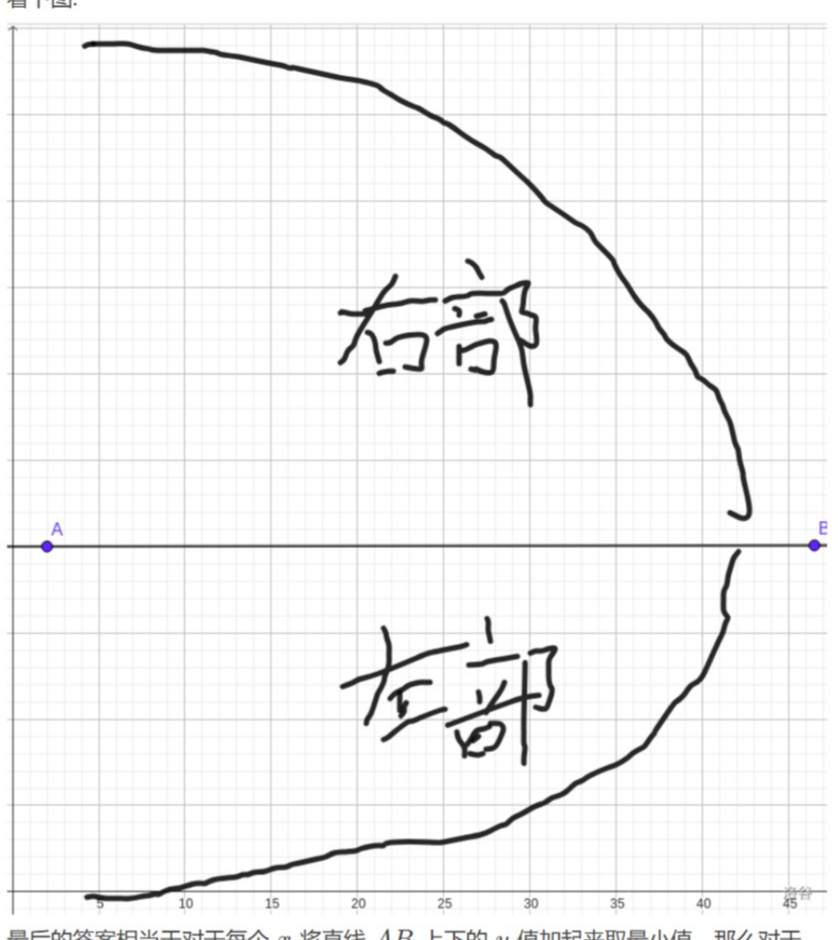
写了以上 Part 3的部分后，就可以拿 $50pts$ 了，可以在联赛 $T4$ 拿到一个不错的分。

Part 4

考虑优化 Part 3.2，首先对于一个叶子节点 i 的 $f_{i,j}$ ，随着 j 从小到大，发现其肯定是贪心地从小到大选代价尽量小的点。
于是 $f_{i,2}-f_{i,1}\le f_{i,3}-f_{i,2}\le f_{i,4}-f_{i,3}...$ ，即 $f_{i,?}$ 关于 j 的函数为一个下凸函数，有了凸性是不是就好做呢?
然而，突如其来的一个代价减去 x_i ，又会使 f_i 丧失凸性甚至是单调性，怎么办呢?
在左部分的求解受阻后，便可以去右部分找点性质，然后如愿以偿地发现右部分是一个上凸函数(若干分段函数的和)，因为和 $n-x$ 有关，将其对个称，就和 x 有关。
然后回到左部，继续从凸性方面优化，大胆地尝试如何消除减 x_i 对凸性的影响，不难想到将所有会影响 f_{i,siz_i} 的加入的点拉低，如下图中的点 I,L ，其中 K 为减去 x_i 后的 f_{i,siz_i} 。



那么这样凸性是有了，但是求出来的 f 肯定是错的，所以，接下来要证明这样操作不影响答案。
看下图:



最后的答案相当于对于每个 x 将直线 AB 上下的 y 值加起来取最小值。那么对于 $f(x)$ 要优于 $f(x+1)$ 当且仅当右部分对应的减少量大于左部分对应的增加量。
那么对于上面那种情况 H,I',L',K ，它们的左部分的增加量一定，右部分的减少量不降，所以最优的要么取到 K 要么取到 H ，中间改变的 I'， L' 不可能取到，所以对最后答案没有影响。
然后，你画个图手玩一下合并后的凸函数，你发现仍然不会影响最后的答案。
具体做法就是个民柯夫斯基和，因为凸包合并的组杂度为 $O(|A|+|B|)$ 直接做的复杂度是还是 $O(n^2)$ ，考虑启发式合并，用平衡树维护斜率，每次先继承重儿子，再将轻儿子的子树节点一个一个插入，减去 x_i 的操作在平衡树上二分最靠前的一个影响凸性的点，然后给这段后缀打一个区间取min的tag即可。
左部分的时间复杂度 $O(nlog^2n)$ ，$(n,m$ 视为同阶)。
这样一来就解决了左边的部分，然后就又可以多拿 $Q=1$ 的 $10pts$ 。
右部分直接做为 nQ，但是发现右部分的上凸包的总边数为 $\sum k$，于是将每条边拿出来在左部分的凸包上二分求出最值即可。
总时间复杂度: $(nlog^2n+\sum klogn)$ ，$(n,m$ 视为同阶)。

Part 4.1

本题还有一种比较自然的做法，考虑每位顾客买花时要使剩余的花的最大值最小，且尽量均匀，形式化的说就是将 s_i 从大到小排序后的字典序最小。
先将所有常客的树建出来，然后从下往上贪心地将区间推平，还是用平衡树+启发式合并维护，然后就相当于将常客都去掉了。
然后加强一下 $m=0$ 的做法，在凸包上二分即可。
感觉这种方法简单很多，而且还不用保证右部分是个上凸壳。
给出第一种做法的代码:

```cpp
#include<bits/stdc++.h>
#define RI int
#define err puts("asd")
#define ll long long
#define ull unsigned long long
#define LL __int128
#define db long double
#define mk make_pair
#define FL fflush(stdout)
#define eb emplace_back
#define FR(u,v) for(int i=h[u],v=a[i].t;i;i=a[i].n,v=a[i].t
#define FB(x,z,y) for(int y=__lg(x&-x)+1,z=x;z;z^=z&-z,y=__
#define FS(x,y) for(int y=x;y;y=(y-1)&x)
#define mem(a,b) memset(a,b,sizeof a)
#define yes puts("Yes")
#define no puts("No")
#define gg puts("-1")
#define vc vector
#define ex exit(0)
#define fi first
#define se second
//#define int long long
//#pragma GCC optimize(2)
//#pragma GCC optimize(3)
using namespace std;

const db eps=1e-15;
const db inf=1e12+5;
const db INF=1e18;
const int mod=1e9+7;

inline ll power(ll x,int y){
    ll t=1;
    while(y){
        if(y&1) t=t*x%mod;
        x=x*x%mod;y>>=1;
    }
    return t;
}

inline void gt(int &x,int &y){if(x>y) swap(x,y);}

inline void cmax(int &x,int y){x<y?x=y:0;}

inline void cmin(db &x,db y){x>y?x=y:0;}

inline void AD(int &x,int y){x+=y;if(x>mod) x-=mod;}

inline ll read(){
    ll s=0;char c=getchar();bool f=0;
    while(c<'0'||c>'9'){if(c=='-') f=1;c=getchar();}
    while(c>='0'&&c<='9') s=(s<<1)+(s<<3)+c-48,c=getchar();
    return f?-s:s;
}

const int N=4e5+5;

struct wu{
    int n,t;
}a[N<<1];

int n,m,K,rt[N],tot,L[N],R[N],C[N],X[N];
int siz[N],pri[N],lc[N],rc[N],cnt,k;
db f[N],val[N],tag[N],sum[N];
int tong[N],stk[N],topf,h[N],p,fa[N],si[N],son[N],D;
pair<int,pair<int,int> >g[N];
vc<int>q[N],V[N];

inline void add(int u,int v){
    a[++p].t=v;a[p].n=h[u];h[u]=p;
}

inline int New(db x,int y=0){
    if(!y) y=++tot;
    else lc[y]=rc[y]=0;
    val[y]=x;f[y]=x;
    tag[y]=inf;siz[y]=1;
    pri[y]=rand();
    return y;
}

inline void up(int x){
    if(!x) return;
    siz[x]=siz[lc[x]]+siz[rc[x]]+1;
    f[x]=f[lc[x]]+f[rc[x]]+val[x];
}

inline void calc(int x,db v){
    f[x]=siz[x]*v;
    cmin(val[x],v);
    cmin(tag[x],v);
}

inline void down(int x){
    if(tag[x]!=inf){
        if(lc[x]) calc(lc[x],tag[x]);
        if(rc[x]) calc(rc[x],tag[x]);
        tag[x]=inf;
    }
}

inline void split(int rt,db k,int &x,int &y){
    if(!rt){x=y=0;return;}
    down(rt);
    if(val[rt]<=k) x=rt,split(rc[rt],k,rc[x],y);
    else y=rt,split(lc[rt],k,x,lc[y]);
    up(rt);
}

inline int merge(int x,int y){
    if(!x||!y) return x^y;
    if(pri[x]>pri[y]){
        down(x);rc[x]=merge(rc[x],y);
        return up(x),x;
    }
    else{
        down(y);lc[y]=merge(x,lc[y]);
        return up(y),y;
    }
}

ll res;
int now;
db vv,S,F;

inline void Do(int x,int &H){
    if(!x) return;
    down(x);
    Do(lc[x],H);Do(rc[x],H);
    RI u,v;
    split(H,val[x],u,v);
    H=merge(merge(u,New(val[x],x)),v);
}

/*inline void out(int x){
    if(!x) return;
    down(x);
    out(lc[x]);
    printf("%.10Lf ",f[x]);
    sb+=val[x];
    out(rc[x]);
}*/

inline void ask(int x){
    if(!x) return;
    down(x);
    if(now!=siz[lc[x]]+1&&vv-f[lc[x]]-val[x]>=val[x]*(now-s
        vv-=f[lc[x]]+val[x];now-=siz[lc[x]]+1;
        S=val[x];ask(rc[x]);
    }
    else ask(lc[x]);
}

inline void solve(int u){
    si[u]=V[u].size();
    FR(u,v){
        solve(v);si[u]+=si[v];
        if(si[v]>si[son[u]]) son[u]=v;
    }
    if(son[u]) rt[u]=rt[son[u]];
    FR(u,v) if(v!=son[u]) Do(rt[v],rt[u]);
    RI x,y;
    for(RI v:V[u]){
        split(rt[u],v,x,y);
        rt[u]=merge(merge(x,New(v)),y);
    }
    if(!X[u]) return;
    S=0;now=siz[rt[u]];vv=f[rt[u]]-X[u];
    if(vv<0){if(rt[u]) calc(rt[u],0);res+=vv;return;}
    ask(rt[u]);
    split(rt[u],S,x,y);
    calc(y,vv/now);
    rt[u]=merge(x,y);
}

inline void getans(int x){
    if(!x) return;
    down(x);
    getans(lc[x]);
    sum[++cnt]=val[x];
    getans(rc[x]);
}

ll P,Q,W;
db ans;

inline db get(int x){
    if(x<W) return INF;
    return sum[x]+P-(x-W)*Q;
}

inline void work(int l,int r){
    if(l>r) return;
    l=n-l;r=n-r;gt(l,r);W=l;
    RI mid,pos;
    while(l<=r){
        mid=l+r>>1;
        if(get(mid)<=get(mid-1)) l=mid+1,pos=mid;
        else r=mid-1;
    }
    ans=min(ans,get(pos));
}

signed main(){
    ll x=0,y=0,z=0,u=0,v=0,SS=0;
    //freopen("12.in","r",stdin);
    //freopen("12.out","w",stdout);
    n=read();m=read();K=read();
    srand(time(0));
    for(RI i=1;i<=n;++i) C[i]=read(),SS+=C[i];
    for(RI i=1;i<=m;++i) L[i]=read(),R[i]=read(),X[i]=read(
    R[++m]=n;q[1].eb(m);
    for(RI i=1;i<=n;++i){
        topf-=tong[i];
        sort(q[i].begin(),q[i].end(),[](int x,int y){return
        for(RI x:q[i]){
            ++tong[R[x]+1];
            if(stk[topf]) add(stk[topf],x);
            else D=x;
            stk[++topf]=x;
        }
        V[stk[topf]].eb(C[i]);
    }
    solve(D);getans(rt[D]);
    for(RI i=1;i<=cnt;++i) sum[i]+=sum[i-1];
    while(K--){
        k=read();z=0;P=Q=0;ans=INF;
        for(RI i=1;i<=k;++i){
            x=read();y=read();
            if(y/x<=n) g[++z]=mk(y/x,mk(y-y/x*x,x)),P+=y;
            else P+=x*n,Q+=x;
        }
        sort(g+1,g+z+1);g[z+1].fi=n;y=z+1;
        for(RI i=z;i>0;--i){
            if(g[i].fi==g[i-1].fi){g[i-1].se.fi+=g[i].se.fi
            work(g[i].fi+1,g[y].fi);
            P-=g[i].se.fi+Q*(g[y].fi-g[i].fi);Q+=g[i].se.se
            y=i;
        }
        work(0,g[1].fi);
        printf("%.0Lf\n",ans+res);
    }
    return 0;
}
```

~~这题数据是真的难造啊!~~