

[比赛链接](#)

题解

G. League of Legends

题意

给定 n 条线段，要求将线段分为 k 组，使得每组的线段交非空，最大化每组的线段交之和。

题解

首先假定所有线没有互相包含的关系，将所有线段 $[l_i, r_i]$ 按 r_i 从大到小排列，设 $\text{dp}(i, j)$ 表示将 j 条线段分到前 i 组得到的答案。

将第 $j+1$ 到 k 条线段分到同一组，如果 $l_k > r_{j+1}$ 则线段交非空且 $\text{dp}(i-1, j)$ 合法，不难得到如下状态转移

$$\text{dp}(i, k) = \text{dp}(i-1, j) + l_k - r_{j+1}$$

由于所有线没有互相包含的关系且 r_i 递减，不难发现 l_i 也递减。

因此对 $i > j$ 如果 $l_k \leq r_{i+1}$ 则一定有 $l_k \leq r_{j+1}$ 同时如果 $l_j \leq r_{k+1}$ 则 $l_i \leq r_{k+1}$ 所以决策具有单调性。

于是每个 i 单调队列维护所有合法 $\text{dp}(i-1, j) - r_{j+1}$ 即可。总时间复杂度 $O(nk)$

接下来考虑某些可以包含其他线段的线段，首先将任何一条线段放入一个已经存在的组一定使得答案不增。

而如果要将一条包含其他线段的线段放入一个已经存在的组，则将它放入一个被它包含的线段所在的组一定使得答案不减，是最佳选择。

因此对满足上述条件的线段只有两种最优选择，一种是放入已经存在的组，这样对答案无贡献。一种是创建一个组，这样贡献为该线段长度。

于是可以处理完所有不包含其他线段的线段，然后枚举他们分成的组数 i 然后取前 $k-i$ 条包含其他线段的线段独立建组构成答案。

至于如果判定一条线段是否包含其他线段，可以将所有线段按 r_i 第一关键字从大到小排序 l_i 第二关键字从小到大排序，然后维护最小右端点。

```
const int MAXN=5e3+5,inf=1e9;
struct Seg{
    int l,r;
    bool operator < (const Seg &b)const{
        return l>b.l||(l==b.l&& r<b.r);
    }
}seg[MAXN],a[MAXN];
int len[MAXN],dp[MAXN][MAXN];
```

```
pair<int,int> que[MAXN];
int main()
{
    int n=read_int(),k=read_int();
    _for(i,0,n){
        seg[i].l=read_int();
        seg[i].r=read_int();
    }
    sort(seg,seg+n);
    int minr=inf,n1=0,n2=0;
    _for(i,0,n){
        if(seg[i].r<minr){
            minr=seg[i].r;
            a[++n1]=seg[i];
        }
        else
            len[++n2]=seg[i].r-seg[i].l;
    }
    mem(dp,-1);
    dp[0][0]=0;
    _rep(i,1,k){
        int head=0,tail=-1;
        _rep(j,1,n1){
            if(~dp[i-1][j-1]){
                pair<int,int> t=make_pair(dp[i-1][j-1]-a[j].l,-a[j].l);
                while(head<=tail&&que[tail]<t)tail--;
                que[++tail]=t;
            }
            while(head<=tail&&a[j].r+que[head].second<=0)head++;
            if(head<=tail)
                dp[i][j]=que[head].first+a[j].r;
        }
    }
    sort(len+1,len+n2+1,greater<int>());
    _rep(i,1,n2)len[i]+=len[i-1];
    int ans=0;
    _rep(i,0,min(n2,k)){
        if(~dp[k-i][n1])
            ans=max(ans,dp[k-i][n1]+len[i]);
    }
    enter(ans);
    return 0;
}
```

赛后总结

jxm过了签到后就一直debug先给自己de然后帮队友de后来de不下去直接重写了一份。再后来就开始罚坐了，甚至没把所有题看完，或许下次应该在罚坐的时候把所有题先看一遍？

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:contest4&rev=1626763611

Last update: 2021/07/20 14:46