肖志昊的20221112模拟考总结

> 你说的对，但是《星战》是由€€£自主研发的一款全旧封闭世界反攻游戏。游戏发生在一个被称作「CSP-S」的OI世界，在这里，被水选中的数据将被授予「全部NO」，导引骗分之力。你将扮演一位名为「总指挥」的神秘角色，在不会的写题中邂逅性格相同、能力一样的数据们，和他们一起获得45pts，找回失散的「YES」——同时，逐步发掘「不可以，总司令」的真相。

上来就甩一个表格~

	神灵庙 (desire)	辉针城 (dealing)	绀珠传 (lunatic)	天空璋 (season)	总分
预估得分	$15pts$	$0pts$	$10pts$	$60pts$	$85pts$
实际得分	$44pts$	$12pts$	$11pts$	$80pts$	$147pts$
满分	$100pts$	$81pts$	$53pts$	$100pts$	$334pts$

为什么实际得分要高这么多？我们首先先分析下这次乱搞の大胜利

分数膨胀——乱搞の大胜利

T1 - 随机化贪心

我们每次加入一个点，会造成一定的代价，每次选择最小代价的决策。

然后发现除了可以过样例以外，用脚造随机数据也可以发现错误。

于是加入随机化，每次概率选择次优解。这竟然能拿 $44pts/81pts$，真的非常离谱。

```
#include<bits/stdc++.h>
#define ll long long
using namespace std;
const int MAXP = 2006110;
const int MAXN = 2006;
int leaf = 1;
int n,a[MAXN];
int q[MAXP][2],tail;
ll solve1(){
    ll ret = 0;
    q[1][0] = a[1],q[1][1] = 0;
    tail = 1;
    for(int i = 2;i <= n;i++){
        int mincost = 0x7fffffff;
        int pos = -1;
        for(int j = 1;j <= tail;j++){
            const int nowv = q[j][0];
            const int nowd = q[j][1];
            if(nowv == -1 || nowd == -1)continue;
            if(nowv + (nowd+2)*a[i] < mincost){
```

```
                mincost = nowv + (nowd+2)*a[i];
                pos = j;
            }
        }
        ret += mincost;
        q[++tail][0] = q[pos][0];
        q[tail][1] = q[pos][1] + 1;
        q[++tail][0] = a[i];
        q[tail][1] = q[pos][1] + 2;
        q[pos][0] = q[pos][1] = -1;
    }
    return ret;
}
ll solve2(int mo){
    ll ret = 0;
    q[1][0] = a[1],q[1][1] = 0;
    tail = 1;
    for(int i = 2;i <= n;i++){
        int mincost = 0x7fffffff;
        int seccost = 0x7fffffff;
        int pos = -1;
        int p2 = -1;
        for(int j = 1;j <= tail;j++){
            const int nowv = q[j][0];
            const int nowd = q[j][1];
            if(nowv == -1 || nowd == -1)continue;
            if(nowv + (nowd+2)*a[i] < mincost){
                seccost = mincost,p2 = pos;
                mincost = nowv + (nowd+2)*a[i];
                pos = j;
            }else if(nowv + (nowd+2)*a[i] < seccost){
                seccost = nowv + (nowd+2)*a[i];
                p2 = j;
            }
        }
        if(rand()% mo == 0 && p2 != -1){
            pos = p2;
            mincost = seccost;
        }
        ret += mincost;
        q[++tail][0] = q[pos][0];
        q[tail][1] = q[pos][1] + 1;
        q[++tail][0] = a[i];
        q[tail][1] = q[pos][1] + 2;
        q[pos][0] = q[pos][1] = -1;
    }
    return ret;
}
int cmp2(int a,int b){
    return a > b;
}
const int MRD = 12;
int rd[MRD] = {5,10,15,20,25,5,5,20,20,36,2,100};
int main(){
    freopen("desire.in","r",stdin);
```

```
    freopen("desire.out","w",stdout);
    srand(time(0));
    scanf("%d",&n);
    for(int i = 1;i <= n;i++){
        scanf("%d",&a[i]);
    }
    sort(a+1,a+n+1,cmp2);
    ll ans = solve1();
    while(clock() < 1800){
        ans = min(ans,solve2(rd[rand()%MRD]));
    }
    printf("%lld",ans);
    return 0;
}
```

T2 - puts("26");

因为如果限制够死，只有 26 种情况，因为 $aaaaaaa...,bbbbbbb...$ 一定是符合条件的，于是 `puts("26");`，竟然能拿 $12pts/81pts$ ，感觉在致敬CCF的Galaxy `puts("NO");`。

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    freopen("dealing.in","r",stdin);
    freopen("dealing.out","w",stdout);
    puts("26");//aaaaa... bbbbbb...
    return 0;
}
```

T3 - 随机化

面对小数据，暴力DFS枚举情况。面对大数据，每次选一个线段，**随机**加入一个集合，统计答案。

就这么简单，朴实无华。朴实地拿了 $11pts/53pts$ ...

```
#include<bits/stdc++.h>
using namespace std;
const int MAXN = 6006;
int n,k,a[MAXN][2],ans = 0;
struct seg{
    int l,r;
}s[MAXN];
void calc(){
    int ret = 0;
    for(int i = 1;i <= k;i++){
        ret += max(0,s[i].r-s[i].l);
    }
    if(ret > ans)ans = ret;
    return ;
}
namespace subtask1{
```

```
    void dfs(int dep){
        if(dep > n){
            calc();
            return ;
        }
        for(int i = 1;i <= k;i++){
            const int orgl = s[i].l;
            const int orgr = s[i].r;
            if(a[dep][0] > s[i].l)s[i].l = a[dep][0];
            if(a[dep][1] < s[i].r)s[i].r = a[dep][1];
            dfs(dep+1);
            s[i].l = orgl,s[i].r = orgr;
        }
        return ;
    }
}
namespace subtask2{
    seg s2[MAXN];
    void pre(){
        for(int i = 1;i <= k;i++){
            s2[i] = s[i];
        }
        return ;
    }
    void solve(){
        for(int i = 1;i <= k;i++){
            s[i] = s2[i];
        }
        for(int i = k+1;i <= n;i++){
            int bel = (rand()%k) + 1;
            if(a[i][0] > s[bel].l)s[bel].l = a[i][0];
            if(a[i][1] < s[bel].r)s[bel].r = a[i][1];
        }
        calc();
    }
    void mainsolve(){
        pre();
        while(clock() < 920){
            solve();
        }
        return ;
    }
}
int main(){
    freopen("lunatic.in","r",stdin);
    freopen("lunatic.out","w",stdout);
    srand(time(0));
    cin>>n>>k;
    for(int i = 1;i <= n;i++){
        cin>>a[i][0]>>a[i][1];
        if(i <= k){
            s[i].l = a[i][0];
            s[i].r = a[i][1];
        }
    }
```

```
    if(n <= 12 || (k <= 4 && n <= 20)){
        subtask1::dfs(k+1);
    }else{
        subtask2::mainsolve();
    }
    cout<<ans;
    return 0;
}
```

T4 - 暴力最小生成树

敲了一个朴素的最小生成树，模拟的过程中加了一个**一维**的差分优化（还不是二维的），理论上来说根据范围能拿 $60pts$ ，但是不知道为什么顶到了 $80pts$ ...

```
#include<bits/stdc++.h>
#define ll long long
using namespace std;
const int MAXN = 5006;
const int MAXM = (5006*5006) / 2;
struct edge{
    int u,v;
}e[MAXM];
ll mp[MAXN][MAXN],n,m,cnt,cost = 0;
bool cmp4edge(edge a,edge b){
    return (mp[a.u][a.v]) < (mp[b.u][b.v]);
}
int fa[MAXN];
int find(int x){
    if(fa[x] == x)return x;
    fa[x] = find(fa[x]);
    return fa[x];
}
int main(){
    freopen("season.in","r",stdin);
    freopen("season.out","w",stdout);
    scanf("%d%d",&n,&m);
    while(m--){
        int a,b,c,d,w;
        scanf("%d%d%d%d%d",&a,&b,&c,&d,&w);
        for(int i = a;i <= b;i++){
            mp[i][c] += w;
            mp[i][d+1] -= w;
        }
    }
    for(int i = 1;i <= n;i++){
        for(int j = 1;j <= n;j++){
            mp[i][j] += mp[i][j-1];
        }
    }
    for(int i = 1;i <= n;i++){
        for(int j = i+1;j <= n;j++){
            e[++cnt].u = i;
            e[cnt].v = j;
```

```
            mp[i][j] += mp[j][i];
        }
    }
    sort(e+1,e+cnt+1,cmp4edge);
    for(int i = 1;i <= n;i++)fa[i] = i;
    for(int i = 1;i <= cnt;i++){
        const int faa = find(e[i].u);
        const int fab = find(e[i].v);
        if(faa != fab){
            fa[fab] = faa;
            cost += mp[e[i].u][e[i].v];
        }
    }
    printf("%lld",cost);
    return 0;
}
```

总结

~~暴力和随机化是神~~

在遇到不会的题的时候要多敲点乱搞的算法，说不定就能拿分了呢。要善用`namespace subtask2`。

传统艺能

放一张哈图上的屑魔女的贴贴图~~（虽然我是沙耶党）~~