

2022.11.12

	预计得分	实际得分	差值
神灵庙	100	24	-76
辉针城	50	15	-35
绀珠传	10	9	-1
天空璋	20	8	-12

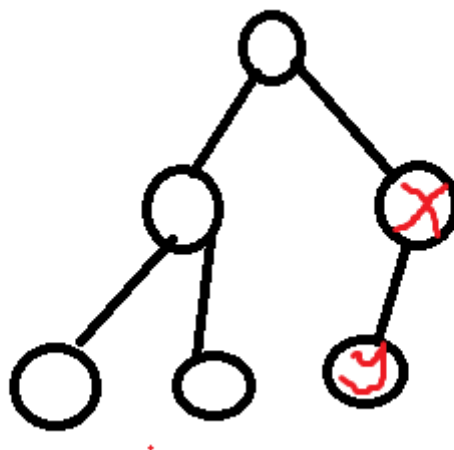
## T1

一眼dz，鉴定为哈夫曼树（明明看着那么像啊啊啊啊），按大小排序，大权值对应小深度，然后寄了（虽然过了两样例

正解是dp

### 分析

首先贪心的思考，如果有



这样一个样式的

树，那么显然删去 $x$ 的子节点 $y$ ，让值挂在 $x$ 上更优，得到一个性质，一个节点要么左右儿子都有，要么没有儿子，那么从当前深度 $d$ 向下就可以转移到 $d+1$ 和 $d+2$ 两个位置

用 $dp[i][d][a][b]$ 表示已经使用前 $i$ 个数，当前深度为 $d$ 的点有 $a$ 个，深度为 $d+1$ 有 $b$ 个，可省略 $d+2$ 那一维，因为相较于下放到 $d+1, d+2$ 明显更劣，且可以在其他深度为 $d+1$ 的节点枚到 $d+2$ （每次会将 $d+1$ 部分转移到下一次的 $a$ 上

设每次使用掉的节点有 $m$ 个，那么就可以转移成

$$dp[i][d][a][b] \rightarrow dp[i+m][d+1][b+a-m][a-m] + d \times (sum[i+m] - sum[i])$$

把深度拆成若干个1相加，则可以优化掉 $d$ 那一维（写一下式子就能推），优化后为：

$$dp[i][a][b] \rightarrow dp[i+m][b+a-m][a-m] + sum[n] - sum[i]$$

但是此时空间降到了 $n^3$ ，时间还是 $O(n^4)$ （因为要多一维 $m$ ），现在考虑优化掉 $m$ ，将刷表法改为填表法，则有

$$dp[i-m][b+m][a-b] + sum[n] - sum[i] \rightarrow dp[i][a][b]$$

我们发现 $(i + m) + (b - m) = i + b$ ，消去了 $m$ 的影响，把常数 $sum[n]$ 单独拿出来，再用一个二维数组来存储剩下的部分，设该数组为 $g$ ，那么： $g[i + b][a - b] = dp[i + m][b - m][a - b] - sum[i]$ ，每次转移 $dp$ 之后，再转移一下 $g$ ： $g[i + a][b] = dp[i][a][b] - sum[i]$ ，就可以优化到跑不满的 $O(n^3)$ 。

(用了整整一上午的时间解决，可恶)

```
#include<bits/stdc++.h>

using namespace std;
typedef long long ll;
ll re(){
    ll s=0,w=1;char c=getchar();
    while(!isdigit(c)){if(c=='-')w=-w;c=getchar();}
    while(isdigit(c)){s=(s<<1)+(s<<3)+(c^48);c=getchar();}
    return s*w;
}
const ll inf=0x3f3f3f3f;
ll INF;
int Min(int x,int y){return x>y?y:x;}
const int D=751;
int n;
int A[D],s[D];
int dp[D][D][D],g[D<<1][D];
signed main()
{
    n=re();
    for(int i=1;i<=n;i++){A[i]=re();}
    sort(A+1,A+1+n,greater<ll>());
    for(int i=1;i<=n;i++){s[i]=s[i-1]+A[i];}
    memset(g,inf,sizeof g);
    g[1][0]=-s[n];
    INF=g[0][0];
    for(int i=0;i<=n;i++){
        for(int a=0;a<=n;a++){
            int up=Min(a,n-a);
            for(int b=0;b<=up;b++){
                if(g[i+b][a-b]==INF)continue;
                dp[i][a][b]=g[i+b][a-b]+s[n];
                g[i+a][b]=Min(g[i+a][b],dp[i][a][b]-s[i]);
            }
        }
    }
    printf("%lld\n",dp[n][0][0]);
    return 0;
}
```

## T2

打了个并查集 $\sum len[i]$ 暴力维护，但是有个小细节写寄了掉大分，暴力加上一些边界条件优化其实能拿到69pts/81pts，终究是错付了

### 正解

很牛逼的倍增并查集，用 $fa[i][k]$ 表示区间 $[i, i + 2^k - 1]$ 集体连到了哪个区间，把查询操作加上一维幂次，找到父亲为0时返回

连边和暴力的并查集差不多，把枚举操作改成倍增操作，联完之后把边传递一波即可

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
ll re(){
    ll s=0,w=1;char c=getchar();
    while(!isdigit(c)){if(c=='-')w=-w;c=getchar();}
    while(isdigit(c)){s=(s<<1)+(s<<3)+(c^48);c=getchar();}
    return s*w;
}
const int D=5e6+514;
const int mod=1e9+7;
ll qp(ll x,ll y){
    ll res=1;
    while(y){
        if(y&1)res=(res*x)%mod;
        x=(x*x)%mod;y>>=1;
    }
    return res;
}
int n,m,lim;
int fa[D][32],num[D];
int len,l,r;
ll ans;
int get(int x,int k){
    return fa[x][k]==0?x:fa[x][k]=get(fa[x][k],k);
}

int main()
{
    n=re();m=re();
    lim=(int)(log(n)/log(2))+1;
    for(int i=1;i<=m;i++){
        len=re();l=re();r=re();
        if(l==r)continue;
        for(int k=lim;k>=0;k--){
            if((len>>k)&1){
                int u=get(l,k);
                int v=get(r,k);
                if(u!=v)fa[u][k]=v;
                l+=1<<k;r+=1<<k;
            }
        }
    }
    for(int j=lim;j>=1;j--){
        for(int i=1;i<=n;i++){
            int x=get(i,j);
            if(x!=i){
                int u=get(i,j-1);int v=get(x,j-1);
                if(u!=v)fa[u][j-1]=v;
                u=get(i+(1<<(j-1)),j-1);v=get(x+(1<<(j-1)),j-1);
                if(u!=v)fa[u][j-1]=v;
            }
        }
    }
    for(int i=1;i<=n;i++)if(get(i,0)==i)ans++;
```

```

ans=(qp(26,ans))%mod;
printf("%11d\n",ans);
return 0;
}

```

## T3

打了个乱搞，先贪心的选择 $k-1$ 条大的单独成组，然后剩下的求一个交集，然后骗了 $9pts/51pts$

正解是在这个基础上先贪心搞一个下界，然后跑dp

可以选择把包含其他线段的线段单独拉出来，剩下的只相交不重合，然后对其进行排序，按左端点升序排列，右端点也是升序，dp转移则是将排序后连续的分组，每段都去求一个交集，单独拿出来的线段为 $x$ 个，一共有 $m$ 组，则剩下的可以分成 $m-x$ 组，用

$dp[i][j] = \max_{k < j} (dp[i-1][k] + \max(0, r_{k+1} - l_i))$ 转移，其中 $l_i$ 是常量（我们枚举到了 $i$ ），就要找出一个 $k$ 使 $r_{k+1}$ 最大就是最优，同时取到0以下的情况可以直接舍弃，所以max也可以去掉；

把一个一个一个一个dp方程写错数列形式，即： $r_{i1} - l_{i2-1} + r_{i2} - l_{i3-1} + r_{i3} \dots$ ，直接把所有 $l_{i-1} + r_i$ 拿出来排序即可，然后求答案

代码还没写出来

## T4

考场上写了个kruskal，暴力的维护 $A[i][j]$ ，然后挂上一棵最小生成树，然后光荣的五颜六色（

正解是一种叫boruvka的东西，我还在学（

## 总·总结

感觉很奇，在考场上推出来dp方程基本是不可能的（对我而言），还有一些奇巧淫技也不是我能想出来的（该类型的题还是做少了），只能说是到要用的时候才知道自己是沙口一个



还是在最后几天多复习一下之前写过的题，多练练板子罢