

Test 20221112

本来不想写这个的，但是毛巨把我卷到了，结果毛巨并没有写多少，然后我也决定水一个

	期望	实际	Δ
T1	100	24	-76
T2	60	60	0
T3	0	3	3
T4	40	40	0
Total	200	127	-73

T1 desire

我是小丑，还以为哈夫曼是正解。

我是小丑。

我是小丑

反正看到这个就想到了 NOI 的那道题 荷马史诗，简单来说就是让这个树的 wpl 最小，而且这道题甚至都不用加 0，岂不美哉？然后就写了个以为是正解的东西，没想到还得是 dp 。

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<algorithm>
#include<queue>
#include<vector>
#define int long long
#define Hangoash
using namespace std;
const int maxn=1e4;
template<typename T>inline void re(T &x)
{
    x=0;
    int f=1;char c=getchar();
    for(;;!isdigit(c);c=getchar())if(c=='-')f=-1;
    for(;;isdigit(c);c=getchar())x=(x<<1)+(x<<3)+(c^48);
    x*=f;
}
template<typename T>inline void wr(T x)
{
    if(x<0)putchar('-'),x=-x;
    if(x>9)wr(x/10);
    putchar(x%10^48);
}
priority_queue<int,vector<int>,greater<int>> >q;
int n,a[maxn];
```

```

inline void pre()
{
    re(n);
    for(register int i=1;i<=n;++i)
        re(a[i]),q.push(a[i]);
}
inline void work(int &ans)
{
    while(q.size()!=1)
    {
        int fir=q.top();q.pop();
        int sec=q.top();q.pop();
        ans+=sec+fir*2;
        q.push(fir+sec);
    }
}
signed main()
{
    #ifdef Hanggoash
    freopen("desire.in","r",stdin);
    freopen("desire.out","w",stdout);
    #endif
    pre();
    if(q.size()==1)return wr(q.top()),0;
    int ans=0;
    work(ans);
    wr(ans);
    return 0;
}

```

T2 dealing

一眼数学题，看到以为是什么容斥计数 dp，结果并没有那么难，但实际上并没有那么简单。

简单地观察了一下发现样例都是 26 的整数次幂，反应过来最后的答案一定是 26 的次方，然后就想到了实际上的答案数是 $[1, n]$ 区间内所有的集合数，每个操作都会把两两之间的数变成一个集合，这明显就可以用并查集来维护了，题目里面还有一个 $\sum k \leq 10^6$ 的部分分，这不就是提醒你暴力维护吗？

然后想了一下正解，觉得一定是倍增或者是用线段树的一些奇葩优化把集合归并的操作从 $O(k)$ 降到 $O(\log k)$ 的，然后到了最后还没想出来。就开下一题了，60pts 实在也没什么其实。

正解就是倍增套在并查集上面。不会

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<algorithm>
#include<queue>
#define int long long
#define Hanggoash
using namespace std;
const int MOD=1e9+7;
const int maxn=1e6+10;
template<typename T>inline void re(T &x)

```

```

{
    x=0;
    int f=1;char c=getchar();
    for(;!isdigit(c);c=getchar())if(c=='-')f=-1;
    for(;isdigit(c);c=getchar())x=(x<<1)+(x<<3)+(c^48);
    x*=f;
}
template<typename T>inline void wr(T x)
{
    if(x<0)putchar('-'),x=-x;
    if(x>9)wr(x/10);
    putchar(x%10^48);
}
inline int power(int a,int b)
{
    int ans=1;
    while(b)
    {
        if(b&1)ans=(ans*a)%MOD;
        a=(a*a)%MOD;
        b>>=1;
    }
    return ans;
}
bool vis[maxn];
int n,m;
int fa[maxn];
inline int find(int x){return x==fa[x]?x:fa[x]=find(fa[x]);}
signed main()
{
    #ifdef Hanggoash
    freopen("dealing.in","r",stdin);
    freopen("dealing.out","w",stdout);
    #endif
    re(n),re(m);
    for(register int i=1;i<=n;++i)fa[i]=i;
    for(register int i=1,Len,x,y;i<=m;++i)
    {
        re(Len),re(x),re(y);
        for(int len=1;len<=Len;len++)
        {
            int fx=find(x+len-1),fy=find(y+len-1);
            if(fx==fy)continue;
            fa[fx]=fy;
        }
    }
    int cnt=0;
    for(register int i=1;i<=n;++i)
    {
        int tmp=find(i);
        if(!vis[tmp])++cnt,vis[tmp]=1;
    }
    wr(power(26,cnt));
    return 0;
}

```

T3 lunatic

这道题是最后来做的，前面三道题的部分分差不多在两个小时左右就写完了，实际上考试的时候觉得部分分应该好拿，但是突破口应该在 T2 和 T3 上面，因为最近做 计数问题 和 线段覆盖 还挺有手感的。

然后就开了这道题，发现一个很 naive 的做法：先把所有线段按照左端点升序排序，这点是毋庸置疑的，然后 $O(n^2)$ 预处理出把编号在 $[l, r]$ 内的线段的交集长度，这样就可以把一个线段交集问题转化为简单地从 n 个数里面划分 k 个连续的段出来然后做一个非常简单的 插板dp 即可的问题，然后就这么写了下去，优化什么的等会再说，定义 $dp[i][j]$ 为插了 i 个板，这个板插在 j 之后的最大值，然后转移也非常显然： $O(n^3)$ 枚举即可，但实际上并没有 de 出来 bug，然后一直写到考试结束也没有找出来，但是却还有 3 分，实在是太扯淡了。

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<algorithm>
#include<queue>
#define Hanggoash
using namespace std;
template<typename T>inline void re(T &x)
{
    x=0;
    int f=1;char c=getchar();
    for(;!isdigit(c);c=getchar())if(c=='-')f=-1;
    for(;isdigit(c);c=getchar())x=(x<<1)+(x<<3)+(c^48);
    x*=f;
}
template<typename T>inline void wr(T x)
{
    if(x<0)putchar('-'),x=-x;
    if(x>9)wr(x/10);
    putchar(x%10^48);
}
const int maxn=1e5+100;
int n,k;
struct Seg{int l,r;}seg[maxn];
bool cmp(Seg a,Seg b){return a.l<b.l;}
int rec[6010][6010];
inline void pre()
{
    re(n),re(k);
    for(register int i=1;i<=n;++i)
        re(seg[i].l),re(seg[i].r);
    sort(seg+1,seg+n+1,cmp);
    for(register int st=1;st<=n;++st)
    {
        int nowl=seg[st].l,nowr=seg[st].r;
        for(register int ed=st,addl,addr;ed<=n;++ed)
        {
            addl=seg[ed].l,addr=seg[ed].r;
            if(addl>=nowr)break;
```

```

        nowl=max(nowl,addl),nowr=min(nowr,addr);
        rec[st][ed]=nowr-nowl+1;
    }
}
}
int dp[6010][6010];
int main()
{
    #ifdef Hanggoash
    freopen("lunatic.in","r",stdin);
    freopen("lunatic.out","w",stdout);
    #endif
    pre();
    for(register int i=1;i<=k-1;++i)
        for(register int j=i;j<=n;++j)
            for(register int ed=0;ed<j;++ed)
                dp[i][j]=max(dp[i][j],dp[i-1][ed]+rec[ed+1][j]);

    int ans=-1;
    for(register int j=1;j<=n;++j)ans=max(ans,dp[k-1][j]);
    wr(ans);
    return 0;
}

```

T4 season

这一眼二维差分然后暴力最小生成树了，写了个 $O(nm + 2n^2)$ 的做法，感觉常数有点偏大了，但也没有办法，毕竟只会这么做了。

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<algorithm>
#include<queue>
#include<unordered_map>
#define int long long
#define Hanggoash
using namespace std;
template<typename T>inline void re(T &x)
{
    x=0;
    int f=1;char c=getchar();
    for(;!isdigit(c);c=getchar())if(c=='-')f=-1;
    for(;isdigit(c);c=getchar())x=(x<<1)+(x<<3)+(c^48);
    x*=f;
}
template<typename T>inline void wr(T x)
{
    if(x<0)putchar('-'),x=-x;
    if(x>9)wr(x/10);
    putchar(x%10^48);
}

```

```

const int maxn=1e7+100;
unordered_map< int,unordered_map<int,int> >sum;
int n,m;
int fa[maxn];
inline int find(int x){return x==fa[x]?x:fa[x]=find(fa[x]);}
inline void pre()
{
    re(n),re(m);
    for(register int i=1;i<=n;++i)fa[i]=i;
    for(register int i=1,a,b,c,d,w;i<=m;++i)
    {
        re(a),re(b),re(c),re(d),re(w);
        for(register int ii=a;ii<=b;++ii)
            sum[ii][c]+=w,sum[ii][d+1]-=w;
    }
    for(register int i=1;i<=n;++i)
        for(register int j=1;j<=n;++j)
            sum[i][j]+=sum[i][j-1];
    for(register int i=1;i<=n;++i)
        for(register int j=1;j<=n;++j)
            sum[i][j]+=sum[i][j-1];
    for(register int j=1;j<=n;++j)
        for(register int i=1;i<=n;++i)
            sum[i][j]+=sum[i-1][j];
}
inline int get(int i,int j){return sum[i][j]-sum[i-1][j]-sum[i][j-1]+sum[i-1][j-1];}
struct Edge{int u,v,w;}E[maxn];int tote;
bool cmp(Edge a,Edge b){return a.w<b.w;}
signed main()
{
    #ifdef Hanggoash
    freopen("season.in","r",stdin);
    freopen("season.out","w",stdout);
    #endif
    pre();
    for(register int i=1;i<=n;++i)
        for(register int j=1;j<=i;++j)
            E[++tote].u=i,E[tote].v=j,E[tote].w=get(i,j)+get(j,i);
    sort(E+1,E+tote+1,cmp);
    int cnt=0,ans=0;
    for(register int i=1;i<=tote;++i)
    {
        if(cnt==n-1)break;
        int x=find(E[i].u),y=find(E[i].v);
        if(x==y)continue;
        ans+=E[i].w;
        fa[x]=y;
        cnt++;
    }
    wr(ans);
    return 0;
}
/*

```

总结

没有总结

硬要说的话感觉就是，太菜了，还要多练，毕竟这次考试该打的部分分都打了，该想的正解也想了，那人菜就是写不出来，还需要找借口吗，直视自己的弱。

哦对，硬要说的话就是 T3 硬刚正解，结果貌似没有写很显而易见的贪心，貌似这个题贪心分还挺高的哈。