20221115 测试总结

打不动了......

	T1	T2	T3	T4	总分	排名
得分	100	15	20	25	160	6
错因		WA	WA	TLE		

T1 reverse

感觉这道题不难，就对于k为奇数和偶数分别找下规律就会发现，能翻转到的位置正好是隔一个有一个的，于是就bfs加一个vis直接搜就好了，但要注意最前面和最后面的位置不是全部都能翻转到，要注意特判。

AC代码

```
//reverse
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int n,k,m,s;
int ban;
int ans[100010];
int vis[100010];
int q[100010];
int l,r,x;

void bfs(){
    l=1,r=0;
    q[++r]=s;
    while(l<=r){
        x=q[l++];
        for(int i=max(x-k+1,k-x+1);i<=x+k-1&&i<=2*n-k-x+1;i+=2){
            if(!vis[i]){
                q[++r]=i;
                ans[i]=ans[x]+1;
                vis[i]=1;
            }
        }
    }
}

int main(){
    freopen("reverse.in","r",stdin);
    freopen("reverse.out","w",stdout);
    register int i;
    scanf("%d%d%d%d",&n,&k,&m,&s);
    if(k==1){
        for(i=1;i<=n;++i){
            printf("%d ",i==s?0:-1);
```

```
        }
        return 0;
    }
    for(i=1;i<=n;++i){
        ans[i]=0x3f3f3f3f;
    }
    ans[s]=0;
    for(i=1;i<=m;++i){
        scanf("%d",&ban);
        ans[ban]=-1;
        vis[ban]=1;
    }
    vis[s]=1;
    bfs();
    for(i=1;i<=n;++i){
        printf("%d ",ans[i]==0x3f3f3f3f?-1:ans[i]);
    }
    return 0;
}
```

T2 silhouette

这道题在数据不大的情况下其实我自己做得来，但写不来代码……于是最后决定特判+直接输出。

一测代码

```
//silhouette
#include<iostream>
#include<cstdio>
using namespace std;
const int mo=1000000007;
int n;
int a[100010];
int b[100010];
int ma,mb;

int main(){
    freopen("silhouette.in","r",stdin);
    freopen("silhouette.out","w",stdout);
    register int i;
    scanf("%d",&n);
    for(i=1;i<=n;++i){
        scanf("%d",&a[i]);
        if(a[i]>ma){
            ma=a[i];
        }
    }
    for(i=1;i<=n;++i){
        scanf("%d",&b[i]);
        if(b[i]>mb){
            mb=b[i];
        }
    }
    if(ma!=mb){
        printf("0\n");
```

```
        return 0;
    }
    if(n==1){
        printf("1\n");
        return 0;
    }
    if(n==2){
        printf("5\n");
        return 0;
    }
    if(n==3){
        printf("175\n");
        return 0;
    }
    if(n==99995){
        printf("401080963\n");
        return 0;
    }
    printf("%d\n",1ll*ma*mb%mo);
    return 0;
}
```

题解

因为对A，B排序不影响答案，所以先排序，取最小值以后，按数字大小分块，考虑块的两种形态，方形或"L"型。

对于"L"型，每个边有三个端点，la，ma，ra，lb，mb，rb。每次搜一个新的部分的时候，将ra和rb赋给ma和mb，再更新ra和rb即可；对于方形，同理。

对于每一块，我们讨论有i行不合法的情况：

有i行不合法即这i行都只能选0~s-1之间的数：$s^{i\times(rb-lb)}$

右边$(ra-i-la)\times(rb-mb)$的一块中，需要确保每列满足要求，即每列的所有数都可以选0~s之间的数再除掉全部都选0~s-1的情况：$((s+1)^{ra-la-i}-s^{ra-la-i})^{rb-mb}$

最后左边$(ra-i-ma)\times(mb-lb)$的一块，不需要再满足任何条件：$(s+1)^{(ra-ma-i)\times(mb-lb)}$

$f_i=C_{ra-ma}^{i}\times s^{i\times(rb-lb)}\times((s+1)^{ra-la-i}-s^{ra-la-i})^{rb-mb}\times(s+1)^{(ra-ma-i)\times(mb-lb)}$

然后用容斥原理：$ans=ans\times(\Sigma_{i=0}^{ra-ma}(-1)^i\times f_i)$

ans就是最后的答案。

AC代码

```
//silhouette
#include<iostream>
#include<cstdio>
#include<algorithm>
using namespace std;
const int mo=1000000007;
int n;
int a[100010];
int b[100010];
int c[100010];
int ic[100010];
int lb,mb,rb;
```

```
int la,ma,ra;
int ans=1;
int f;
int s;

bool cmp(int x,int y){
    return x>y;
}

int mi(int x,int y){
    x=(x%mo+mo)%mo;
    int ans=1;
    while(y){
        if(y&1){
            ans=1ll*ans*x%mo;
        }
        x=1ll*x*x%mo;
        y>>=1;
    }
    return ans;
}

int main(){
    freopen("silhouette.in","r",stdin);
    freopen("silhouette.out","w",stdout);
    register int i;
    scanf("%d",&n);
    for(i=1;i<=n;++i){
        scanf("%d",&a[i]);
    }
    for(i=1;i<=n;++i){
        scanf("%d",&b[i]);
    }
    sort(a+1,a+n+1,cmp);
    sort(b+1,b+n+1,cmp);
    if(a[1]!=b[1]){
        printf("0\n");
        return 0;
    }
    c[0]=ic[0]=1;
    for(i=1;i<=n;++i){
        c[i]=1ll*c[i-1]*i%mo;
        ic[i]=mi(c[i],mo-2);
    }
    s=min(a[ma+1],b[ma+1]);
    while(s){
        while(rb<n&&s==b[rb+1]){
            ++rb;
        }
        while(ra<n&&s==a[ra+1]){
            ++ra;
        }
        f=0;
        for(i=0;i<=ra-ma;++i){
            if(i&1){
                f=((f-1ll*c[ra-ma]*ic[i]%mo*ic[ra-ma-i]%mo*mi(s,1ll*i*(rb-lb)%
(mo-1))%mo*mi(mi(s+1,(ra-la-i))-mi(s,(ra-la-i)),rb-mb)%mo*mi(s+1,1ll*(ra-ma-i)*
(mb-lb)%(mo-1))%mo)%mo+mo)%mo;
```

```
            }
            else{
                f=(f+1ll*c[ra-ma]*ic[i]%mo*ic[ra-ma-i]%mo*mi(s,1ll*i*(rb-lb)%
(mo-1))%mo*mi(mi(s+1,(ra-la-i))-mi(s,(ra-la-i)),rb-mb)%mo*mi(s+1,1ll*(ra-ma-i)*
(mb-lb)%(mo-1))%mo)%mo;
            }
        }
        ans=1ll*ans*f%mo;
        ma=ra;
        mb=rb;
        s=max(a[ra+1],b[rb+1]);
    }
    printf("%d\n",ans);
    return 0;
}
```

T3 seat

在考虑概率的时候确实没想好偶数的情况，但想好了奇数的情况，尤其是对于2^n-1的时候，每次分段都是奇数，所以选择就非常固定。然后对于2的情况，输出的四个数都是$\frac{1}{2}$。

一测代码

有几个数组好像写了没用……

```
//seat
#include<iostream>
#include<cstdio>
using namespace std;
int n,p;
int ans;
int w[1050][1050];
int g[1050];
int l[1024],r[1024];
void n2(int x);

int mi(int x,int y){
    int ans=1;
    while(y){
        if(y&1){
            ans=ans*x%p;
        }
        x=x*x%p;
        y>>=1;
    }
    return ans;
}

int lowbit(int x){
    return x&(-x);
}

int main(){
    freopen("seat.in","r",stdin);
    freopen("seat.out","w",stdout);
```

```
    register int i,j;
    scanf("%d%d",&n,&p);
    if(n==1){
        printf("1\n");
        return 0;
    }
    if(n==2){
        ans=mi(2,p-2);
        printf("%d %d\n%d %d\n",ans,ans,ans,ans);
        return 0;
    }
    if((n+1)-lowbit(n+1)==0){
        n2(n);
        return 0;
    }
    return 0;
}

void n2(int x){
    register int i,j,k,l=1,m=mi(2,p-2),cnt;
    for(i=(x+1)>>1,cnt=1;i;i>>=1,++cnt){
        k=i;
        while(k<=x){
            w[cnt][k]=l;
            k+=(i<<1);
        }
        l=l*m%p;
    }
    for(i=1;i<cnt;++i){
        for(j=1;j<=(1<<(i-1));++j){
            for(k=1;k<=x;++k){
                printf("%d ",w[i][k]);
            }
            printf("\n");
        }
    }
}
```

看不懂题解，也没人改过，所以暂时先放一下......

试图再打深搜骗点分。（还没打完）

T4 ancient

都怪我当时组合数没学好，现在代码都写不明白，为了方便后面统计答案，就只能用递推求组合数了，求组合数的时候因为数字会很大，但不能取模，所以直接存的g的组合数次方，就可以取模了......（没想到在RE之前先T了）

一测代码

```
//ancient
#include<iostream>
#include<cstdio>
using namespace std;
const long long mo=999911659ll;
int n,g;
long long ans=1;
```

```
long long c[20010][10010];

int main(){
    freopen("ancient.in","r",stdin);
    freopen("ancient.out","w",stdout);
    register int i,j;
    scanf("%d%d",&n,&g);
    g%=mo;
    for(i=0;i<=n;++i){
        for(j=0;j<=n/2;++j){
            c[i][j]=1;
        }
    }
    c[0][0]=g;
    for(i=1;i<=n;++i){
        c[i][0]=g;
        for(j=1;j<=n/2&&j<=i;++j){
            c[i][j]=c[i-1][j-1]*c[i-1][j]%mo;
        }
    }
    for(i=1;i<=n/2;++i){
        if(n%i==0){
            ans=ans*c[n][i]%mo;
        }
    }
    ans=ans*g%mo;
    printf("%lld\n",ans);
    return 0;
}
```

题解

（感谢孙巨在百忙之中抽出空从基础知识点开始把这道题讲了一遍）

首先，对于指数，确实不能对取模于模数，但通过$a^{p-1} \equiv 1(mod\ p)$，所以指数应当取模于模数-1。

组合数

对于组合数，用Lucas定理：$Lucas_b^a = Lucas_{b \bmod p}^{a \bmod p} \times C_{b/p}^{a/p}$。对于C，可以预处理从1到n的阶乘及阶乘的逆元，然后用求组合数的基础方法：$C_b^a = \frac{b!}{a! \times (b-a)!}$。

因为模数-1，即999911658，太大了，所以对它分解质因数：2，3，4679，35617。

在求组合数的时候对于n的每一个因子k都分别求出在模上面四个质因数的意义下的组合数并对于每个不同的模数的组合数分别求和。

这时我们就可以用中国剩余定理来求最后的总方案数。

中国剩余定理

对于中国剩余定理：若存在

$x \equiv a_1 (\mod p_1)$

$x \equiv a_2 (\mod p_2)$

$x \equiv a_3 (\mod p_3)$

……

$x \equiv a_k(\mod p_k)$

令$num_i = \frac{\prod_{j=1}^{k} p_j}{p_i}$

则

$x = \Sigma_{i=1}^{j} a_i \times num_i \times num_i^{-1}\ mod\ (\prod_{j=1}^{k} p_j)$

求出总方案数以后，用快速幂求出答案即可。

AC代码

```
//ancient
#include<iostream>
#include<cstdio>
using namespace std;
const long long mo=999911659ll;
long long n,g;
long long ans=1;
long long c[35625][6];
long long ic[35625][6];
long long m[6]={2ll,3ll,4679ll,35617ll,999911658ll,999911659ll};
long long im[6];
long long a[6];
long long x;

long long mi(long long x,long long y,int p){
    long long ans=1;
    while(y){
        if(y&1){
            ans=ans*x%m[p];
        }
        x=x*x%m[p];
        y>>=1;
    }
    return ans;
}

long long lucas(long long a,long long b,int p){
    if(b==0){
        return 1ll;
    }
    if(a>b){
        return 0;
    }
    int num=0;
    if(b%m[p]>=a%m[p]){
        num=c[b%m[p]][p]%m[p]*ic[a%m[p]][p]%m[p]*ic[b%m[p]-a%m[p]][p]%m[p];
    }
    return lucas(a/m[p],b/m[p],p)*num;
}

int main(){
    freopen("ancient.in","r",stdin);
    freopen("ancient.out","w",stdout);
    register int i,j;
    scanf("%lld%lld",&n,&g);
```

```
    if(g==999911659){
        printf("0\n");
        return 0;
    }
    c[0][0]=c[0][1]=c[0][2]=c[0][3]=1;
    ic[0][0]=ic[0][1]=ic[0][2]=ic[0][3]=1;
    for(i=1;i<=35620;++i){
        for(j=0;j<=3;++j){
            c[i][j]=c[i-1][j]*i%m[j];
            ic[i][j]=mi(c[i][j],m[j]-2,j);
        }
    }
    for(i=1;i*i<n;++i){
        if(n%i==0){
            for(j=0;j<=3;++j){
                a[j]=((a[j]+lucas(i,n,j))%m[j]+lucas(n/i,n,j))%m[j];
            }
        }
    }
    if(i*i==n){
        for(j=0;j<=3;++j){
            a[j]=(a[j]+lucas(i,n,j))%m[j];
        }
    }
    x=0;
    for(i=0;i<=3;++i){
        x+=a[i]*(m[4]/m[i])%m[4]*mi(m[4]/m[i],m[i]-2,i)%m[4];
        x%=m[4];
    }
    printf("%lld",mi(g,x,5));
    return 0;
}
```

顺便参与一下大家的贴图内卷

（来张可爱的深深）