

牛客多校第六场

数学题较多。

B

太菜了。

```
#include<stdio.h>

#define mod 1000000007

long long a[20000010]; //函数值
long long b[20000010]; //异或和

void init()
{
    b[0]=0;
    a[0]=1;
    long long temp5=1;
    int i;
    for(i=1;i<20000005;i++)
    {
        temp5*=500000004;
        temp5%=1000000007;
        a[i]=((a[i-1]*(mod-temp5+1))%mod);
        b[i]=b[i-1]^a[i];
    }
}

int main()
{
    init();
    int T;
    scanf("%d",&T);
    while(T--)
    {
        int p;
        scanf("%d",&p);
        printf("%lld\n",b[p]);
    }
}
```

C

水。

```
#include<stdio.h>

long long int n,m,t;
double b[205],ans[205],c[205][205],a[205][205],ans1;

int main()
{
    scanf("%lld",&t);
    while(t--)
    {
        scanf("%lld%lld",&n,&m);
        long long i,j;
        for(i=0;i<n;i++)
        {
            for(j=0;j<m;j++)
            {
                scanf("%lf",&a[i][j]);
            }
        }
        ans1=0;
        for(j=0;j<m;j++)
        {
            b[j]=0;
            ans[j]=0;
            for(i=0;i<n;i++)
            {
                b[j]+=a[i][j];
                c[i][j]=b[j]/a[i][j];
                ans[j]=(ans[j]>c[i][j])?ans[j]:c[i][j];
            }
            ans1=(ans1>ans[j]?ans1:ans[j]);
        }
        printf("%.8f\n",ans1);
    }
}
```

E

构造很有趣，有一点难度。不过在下面直接给出来了，应该不用多讲吧。

```
#include<stdio.h>
```

```
int n,k,t;

int main()
{
    scanf("%d%d",&n,&k);
    if(n%2)
    {
        if(k!=0)
        {
            printf("-1\n");
        }
        else
        {
            t=n;
            int cnt=0;
            while(t--)
            {
                if(t%2)
                {
                    printf("%d ",cnt);
                }
                else
                {
                    printf("%d ",n-cnt);
                    cnt++;
                }
            }
        }
    }
    else
    {
        if(k!=n/2)
        {
            printf("-1\n");
        }
        else
        {
            printf("%d %d ",n,n/2);
            t=n-2;
            int cnt=1;
            while(t--)
            {
                if(t%2)
                {
                    printf("%d ",cnt);
                }
                else
                {
                    printf("%d ",n-cnt);
                    cnt++;
                }
            }
        }
    }
}
```

```
}  
}  
}  
}
```

F

以下是补题。

G

纯粹的构造题。隔壁的天才题解告诉我们，循环填入就完事了。啊，我为什么就没想到呢.....

```
#include<stdio.h>  
  
int r[205][205],c[205][205];  
  
int main()  
{  
    int t;  
    scanf("%d",&t);  
    while(t--)  
    {  
        int n,k;  
        scanf("%d%d",&n,&k);  
        if(k==1||n==1||2*(n+1)*n%k)  
        {  
            printf("-1\n");  
            continue;  
        }  
        int res=0;  
        int i;  
        for(i=1;i<=n;i++)  
        {  
            int j;  
            for(j=1;j<=n;j++)  
            {  
                r[i][j]=res;  
                res=(res+1)%k;  
            }  
            for(j=1;j<=n+1;j++)  
            {  
                c[j][i]=res;  
            }  
        }  
    }  
}
```

```

        res=(res+1)%k;
    }
}
for(i=1;i<=n;i++)
{
    r[n+1][i]=res;
    res=(res+1)%k;
}
for(i=1;i<=n+1;i++)
{
    int j;
    for(j=1;j<=n;j++)
    {
        printf("%d ",c[i][j]+1);
    }
    puts("");
}
for(i=1;i<=n+1;i++)
{
    int j;
    for(j=1;j<=n;j++)
    {
        printf("%d ",r[i][j]+1);
    }
    puts("");
}
}
return 0;
}

```

H

传说中的数位DP——标程如下。

```

#include<stdio.h>
#include<string.h>

void Add(int *x,int y)
{
    (*x)+=y;
    if((*x)>=1000000007)
    {
        (*x)-=1000000007;
    }
}

int dp[105][2005][2][2];
char s[105];

```

```
int n,v[105];

int main()
{
    scanf("%s",s+1);
    n=strlen(s+1);
    int i;
    for(i=1;i<=n;i++)
    {
        v[i]=s[i]-'0';
    }
    dp[0][1000][0][0]=1;
    for(i=0;i<n;i++)
    {
        int d;
        for(d=0;d<2005;d++)
        {
            int f0;
            for(f0=0;f0<2;f0++)
            {
                int f1;
                for(f1=0;f1<2;f1++)
                {
                    if(!dp[i][d][f0][f1])
                    {
                        continue;
                    }
                    int V=v[i+1];
                    int A;
                    for(A=0;A<10;A++)
                    {
                        int B;
                        for(B=0;B<10;B++)
                        {
                            if(A>B&&!f0)
                            {
                                continue;
                            }
                            if(B>V&&!f1)
                            {
                                continue;
                            }
                            int nf0=f0|(A<B);
                            int nf1=f1|(B<V);
                            Add(&dp[i+1][d+A-
B][nf0][nf1],dp[i][d][f0][f1]);
                        }
                    }
                }
            }
        }
    }
}
```

```

    }
}
int res=0;
int d;
for(d=1000+1;d<2005;d++)
{
    int f1;
    for(f1=0;f1<2;f1++)
    {
        Add(&res,dp[n][d][1][f1]);
    }
}
printf("%d\n",res);
return 0;
}

```

I

Stling数

```

#include<stdio.h>

int MOD;

long long powmod(long long x,int k)
{
    long long ans=1;
    while(k)
    {
        if(k&1)
        {
            ans=ans*x%MOD;
        }
        x=x*x%MOD;
        k>>=1;
    }
    return ans;
}

int getfact(int x,int *p)
{
    int t=x,sz=0;
    int i;
    for(i=2;i*i<=t;i++)
    {
        if(x%i==0)
        {
            p[++sz]=i;
        }
    }
}

```

```
        while(x%i==0)
        {
            x/=i;
        }
    }
}
if(x>1)
{
    p[++sz]=x;
}
return sz;
}

long long facd[1000005], facv[1000005];
long long G, mi[1000005], inv[1000005];

void pre()
{
    facd[0]=1;
    int i;
    for(i=1; i<MOD; i++)
    {
        facd[i]=facd[i-1]*i%MOD;
    }
    facv[MOD-1]=facd[MOD-1];
    for(i=MOD-2; i>=0; i--)
    {
        facv[i]=facv[i+1]*(i+1)%MOD;
    }
    int prime[10];
    int sz=getfact(MOD-1, prime);
    for(G=1; G++;)
    {
        bool ok=1;
        for(i=1; i<=sz; i++)
        {
            if(powmod(G, (MOD-1)/prime[i])==1)
            {
                ok=0;
                break;
            }
        }
        if(ok)
        {
            break;
        }
    }
    mi[0]=1;
    for(i=1; i<MOD-1; i++)
    {
```

```

        mi[i]=mi[i-1]*G%MOD;
    }
    inv[1]=1;
    for(i=2;i<MOD;i++)
    {
        inv[i]=(MOD-MOD/i)*inv[MOD%i]%MOD;
    }
}

long long C(long long n,long long m)
{
    return (n<m)?0:facd[n]*facv[m]%MOD*facv[n-m]%MOD;
}

long long calc(long long n,long long m)
{
    if(!m)
    {
        return 1;
    }
    if(n<m)
    {
        return 0;
    }
    return C(n%MOD,m%MOD)*calc(n/MOD,m/MOD)%MOD;
}

long long query(long long n,long long m)
{
    m=(n<m)?n:m;
    long long s=0;
    int i;
    for(i=0;i<MOD-1;i++)
    {
        int x=mi[i];
        if(x+n>MOD)
        {
            continue;
        }
        if(!i)
        {
            s=(s+(m+1)*facd[x+n-1]%MOD*facv[x-1])%MOD;
        }
        else
        {
            s=(s+(mi[(MOD-1-i*(m+1)%(MOD-1))%(MOD-1)]-1LL)*inv[mi[MOD-1-i-1LL]%MOD]*facd[x+n-1]%MOD*facv[x-1])%MOD;
        }
    }
    s=(MOD-s)%MOD;
    if(n==MOD-1)

```

```
{
    s=(s-1LL+MOD)%MOD;
}
return s;
}

long long solve(long long n,long long m)
{
    long long u1=n/MOD,v1=n%MOD;
    if(m<u1)
    {
        return 0;
    }
    m-=u1;
    long long u2=m/(MOD-1),v2=m%(MOD-1);
    long long s=0;
    if(u2)
    {
        s=(s+(((u2-1)&1)?MOD-1:1)*calc(u1-1,u2-1)%MOD*query(v1,MOD-2))%MOD;
    }
    s=(s+((u2&1)?MOD-1:1)*calc(u1,u2)%MOD*query(v1,v2))%MOD;
    if(v1==MOD-1&&u2)
    {
        s=(s+(((u2-1)&1)?MOD-1:1)*calc(u1-1,u2-1))%MOD;
    }
    return s*((u1&1)?MOD-1:1)%MOD;
}

int main()
{
    long long n,l,r;
    scanf("%lld%lld%lld%d",&n,&l,&r,&MOD);
    pre();
    long long s1=solve(n,r);
    long long s2=solve(n,l-1);
    printf("%lld\n",(s1-s2+MOD)%MOD);
    return 0;
}
```

K

最初我开了两个map[]结果TLE[]

这个版本的标程里用到了stable_sort[](例如归并排序)，亲测删掉stable_则通过率0%。因此需要保证排序时元素有序性。

```
#include<stdio.h>
```

```
#include<algorithm>
#include<vector>

using namespace std;

int a[500111],ord[500111],n,k;

bool cmp(int x,int y)
{
    return a[x]<a[y];
}

bool check()
{
    int i;
    for(i=1;i<=n;i++)
    {
        ord[i]=i;
    }
    stable_sort(ord+1,ord+n+1,cmp);
    vector<pair<int,int> > vs;
    for(i=2;i<=n;i++)
    {
        if(a[ord[i]]==a[ord[i-1]])
        {
            int p1=ord[i-1],p2=ord[i];
            if(p2-p1>=k)
            {
                continue;
            }
            int vl=p2%k,vr=(p1-1)%k;
            if(vl<=vr)
            {
                vs.push_back(make_pair(vl,vr));
            }
            else
            {
                vs.push_back(make_pair(vl,k-1));
                vs.push_back(make_pair(0,vr));
            }
        }
    }
    sort(vs.begin(),vs.end());
    int MX=-1;
    for(i=0;i<(int)vs.size();i++)
    {
        if(MX<vs[i].first-1)
        {
            return true;
        }
        MX=(MX>vs[i].second?MX:vs[i].second);
    }
}
```

```
    }
    if(MX<k-1)
    {
        return true;
    }
    return false;
}

void solve()
{
    scanf("%d%d",&n,&k);
    int i;
    for(i=1;i<=n;i++)
    {
        scanf("%d",&a[i]);
    }
    if(check())
    {
        puts("YES");
    }
    else
    {
        puts("NO");
    }
}

int main()
{
    int tc;
    scanf("%d",&tc);
    while(tc--)
    {
        solve();
    }
    return 0;
}
```

From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: <https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:%E7%89%9B%E5%AE%A2%E5%A4%9A%E6%A0%A1%E7%AC%AC%E5%85%AD%E5%9C%BA&rev=1596026210>

Last update: 2020/07/29 20:36