

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。

本题用到了斐波那契数和卢卡斯数的结论。 [斐波那契和卢卡斯](#)

```
#include<stdio.h>  
  
#include<algorithm>  
#include<set>  
  
using namespace std;  
  
set<pair<int,int> > s;  
  
long long L[44],ans;  
  
void Sub(set<pair<int,int> >::iterator p)  
{  
    set<pair<int,int> >::iterator pl=p,pr=p;  
    int l=(*p).second,r=(*p).first,last,plast;  
    if(p==s.begin())  
    {  
        ans-=l>>1;  
        if(l<=3)  
        {  
            ans-=r-l>>1;  
            plast=r;  
        }  
        else  
        {  
            ans-=r-l;  
            plast=r-1;  
        }  
        last=1;  
    }  
    else  
    {  
        --pl;  
        last;  
        if((*pl).second<=3)  
        {  
            last=(*pl).first;
```

```
    }
    else
    {
        last=(*pl).first-1;
    }
    ans--=(l-last+1)/2;
    ans--=r-l;
    plast=r-1;
}
++pr;
if(pr==s.end())
{
    return;
}
ans+=((*pr).second-last+1)>>1;
ans-=((*pr).second-plast+1)>>1;
return;
}

void Add(set<pair<int,int> >::iterator p)
{
    set<pair<int,int> >::iterator pl=p,pr=p;
    int l=(*p).second,r=(*p).first,last,plast;
    if(p==s.begin())
    {
        ans+=l>>1;
        if(l<=3)
        {
            ans+=r-l>>1;
            plast=r;
        }
        else
        {
            ans+=r-l;
            plast=r-1;
        }
        last=1;
    }
    else
    {
        --pl;
        last;
        if((*pl).second<=3)
        {
            last=(*pl).first;
        }
        else
        {
            last=(*pl).first-1;
        }
        ans+=((l-last+1)/2);
    }
}
```

```
        ans+=r-l;
        plast=r-1;
    }
    ++pr;
    if(pr==s.end())
    {
        return;
    }
    ans-=((*pr).second-last+1)>>1;
    ans+=((*pr).second-plast+1)>>1;
    return;
}

void add(int x)
{
    if(!x)
    {
        return;
    }
    if(x==1)
    {
        x=2;
    }
    set<pair<int,int> >::iterator p=s.lower_bound(make_pair(x-1,-1));
    if(p==s.end()||(*p).second>x+1)
    {
        int l=(1ll<<31)-1,r;
        if(p!=s.end())
        {
            l=(*p).second;
            r=(*p).first;
        }
        if(l==x+2)
        {
            Sub(p);
            s.erase(p);
        }
        else
        {
            r=x;
        }
        p=s.lower_bound(make_pair(x-2,-1));
        if(p!=s.end()&&(*p).first==x-2)
        {
            l=(*p).second;
            Sub(p);
            s.erase(p);
        }
        else
        {
```

```

        l=x;
    }
    s.insert(make_pair(r,l));
    Add(s.lower_bound(make_pair(r,l)));
    return;
}
if((*p).first==x-1)
{
    int l=(*p).second;
    Sub(p);
    s.erase(p);
    if(l<x-1)
    {
        s.insert(make_pair(x-3,l));
        Add(s.lower_bound(make_pair(x-3,l)));
    }
    add(x+1);
    return;
}
if((( *p).first-x)&1)
{
    int l=(*p).second,r=(*p).first;
    Sub(p);
    s.erase(p);
    if(l<x)
    {
        s.insert(make_pair(x-1,l));
        Add(s.lower_bound(make_pair(x-1,l)));
    }
    l=r+1,r=r+1;
    p=s.lower_bound(make_pair(r+2,-1));
    if(p!=s.end()&&(*p).second==r+2)
    {
        r=(*p).first;
        Sub(p);
        s.erase(p);
    }
    s.insert(make_pair(r,l));
    Add(s.lower_bound(make_pair(r,l)));
    return;
}
else
{
    int l=(*p).second,r=(*p).first;
    Sub(p);
    s.erase(p);
    if(l<x)
    {
        s.insert(make_pair(x-1,l+1));
        Add(s.lower_bound(make_pair(x-1,l+1)));
    }
}

```

```
int nl=r+1;
r=r+1;
p=s.lower_bound(make_pair(r+2,-1));
if(p!=s.end()&&>(*p).second==r+2)
{
    r>(*p).first;
    Sub(p);
    s.erase(p);
}
p=s.lower_bound(make_pair(nl-2,-1));
if(p!=s.end()&&>(*p).first==nl-2)
{
    nl>(*p).second;
    Sub(p);
    s.erase(p);
}
s.insert(make_pair(r,nl));
Add(s.lower_bound(make_pair(r,nl)));
add(l-2);
return;
}
}

void del(int x)
{
    if(!x)
    {
        return;
    }
    if(x==1)
    {
        x=2;
    }
    set<pair<int,int> >::iterator p=s.lower_bound(make_pair(x,-1));
    if((*p).second>x)
    {
        int r>(*p).first,l>(*p).second;
        Sub(p);
        s.erase(p);
        if(l<r)
        {
            s.insert(make_pair(r,l+2));
            Add(s.lower_bound(make_pair(r,l+2)));
        }
        if((x-l)&1)
        {
            if(l-2>=x+1)
            {
                p=s.lower_bound(make_pair(x-1,-1));
                if(p==s.end()||(*p).first!=x-1)

```

```

        {
            s.insert(make_pair(l-2,x+1));
            Add(s.lower_bound(make_pair(l-2,x+1)));
        }
        else
        {
            int nl=(*p).second;
            Sub(p);
            s.erase(p);
            s.insert(make_pair(l-2,nl));
            Add(s.lower_bound(make_pair(l-2,nl)));
        }
    }
    add(l-2);
}
else
{
    p=s.lower_bound(make_pair(x-1,-1));
    if(p==s.end()||(*p).first!=x-1)
    {
        s.insert(make_pair(l-1,x+1));
        Add(s.lower_bound(make_pair(l-1,x+1)));
    }
    else
    {
        int nl=(*p).second;
        Sub(p);
        s.erase(p);
        s.insert(make_pair(l-1,nl));
        Add(s.lower_bound(make_pair(l-1,nl)));
    }
}
return;
}
if(((*p).second-x)&1)
{
    int r=(*p).first,l=(*p).second;
    Sub(p);
    s.erase(p);
    s.insert(make_pair(x-1,l));
    Add(s.lower_bound(make_pair(x-1,l)));
    if(x+3<=r)
    {
        s.insert(make_pair(r,x+3));
        Add(s.lower_bound(make_pair(r,x+3)));
    }
    add(x-1);
    return;
}
else
{

```

```
int r>(*p).first,l>(*p).second;
Sub(p);
s.erase(p);
if(r>x)
{
    s.insert(make_pair(r,x+2));
    Add(s.lower_bound(make_pair(r,x+2)));
}
if(l<x)
{
    s.insert(make_pair(x-2,l));
    Add(s.lower_bound(make_pair(x-2,l)));
}
return;
}
}

int X[1000],Y[1000];

void solve()
{
    int a,b;
    scanf("%d%d",&a,&b);
    ++b;
    int op=0,OP=0;
    if(a<0)
    {
        OP=1;
        a=-a;
    }
    int cnt=0,cnt2=0;
    while(a)
    {
        int k=upper_bound(L+1,L+44,a)-L-1;
        a-=L[k];
        op=OP;
        if(op&1)
        {
            Y[cnt2++]=b+k;
        }
        else
        {
            X[cnt++]=b+k;
        }
        if(b-k>=0)
        {
            op^=k&1;
            if(op&1)
            {
                Y[cnt2++]=b-k;
            }
        }
    }
}
```

```
        }
        else
        {
            X[cnt++] = b - k;
        }
    }
    else
    {
        op ^= k & 1;
        if ((b - k + 1) & 1)
        {
            op ^= 1;
        }
        if (op & 1)
        {
            Y[cnt2++] = k - b;
        }
        else
        {
            X[cnt++] = k - b;
        }
    }
}
int i;
for (i = 0; i < cnt; ++i)
{
    add(X[i]);
}
for (i = 0; i < cnt2; ++i)
{
    del(Y[i]);
}
printf("%lld\n", ans);
}

int main()
{
    L[0] = 2;
    L[1] = 1;
    int i;
    for (i = 2; i < 44; ++i)
    {
        L[i] = L[i - 1] + L[i - 2];
    }
    int T;
    scanf("%d", &T);
    while (T--)
    {
        ans = 0;
        s.clear();
        int n;
```

```
scanf("%d",&n);  
while(n--)  
{  
    solve();  
}  
}
```

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

题目、解答与代码见[stirling数](#)

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%98%E5%AE%A2%E5%A4%9A%E6%A0%A1%E7%AC%AC%E5%85%AD%E5%9C%BA&rev=1596248144>

Last update: 2020/08/01 10:15