牛客多校第五场

这场的题目好啊！

一般有规律，奇数场可能偏向数学计算，偶数场更考验~~玛俐~~码力。在本人的码力实在不足的情况下，看来更喜欢考验思维量的题目。

（成功地首次独立写出了Prim算法/*优先队列优化*/——然而什么也没有发生/*只过了10%样例*/）

D

因为轮换不影响，不妨在纸上按照圆形排序整个序列，找规律会发现其实就是一个简单的最长上升子序列DP就完事了。

（和弦转位的背景还挺有趣的）

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

int j, s, n, t, a[100001], b[100001], ans = 0;

int main()
{
    scanf("%d",&n);
    a[0]=-1000000;
    int i;
    for(i=1;i<=n;i++)
    {
        scanf("%d", &b[i]);
        b[i + 2 * n] = b[i + n] = b[i];
    }
    int j;
    for(j = 0; j < n; j++)
    {
        s = 0;
        memset(a, 0, sizeof(a));
        for(i = 0 + j; i < n + j; i++)
        {
            t = b[i + 1];
            if(t > a[s]) a[++s] = t;
            else
            {
                int l = 1, h = s, m;
                while(l <= h)
                {
                    m = (l + h) / 2;
                    (t > a[m])?(l = m + 1):(h = m - 1);
                }
                a[l] = t;
```

```
            }
        }
        ans =(ans>s)?ans:s;
    }
    printf("%d\n",n-ans);
}
```

E

我们队是用Java过的。本来想写个高精度，~~奈何不会写，~~奈何有这样取巧的方法。

我还考虑要不要存入set的时候保证互素，使得高精度只需FFT就行了——后来发现Java竟然可以完美避过。马佬看来之前用过BigInteger类，我还是第一次听说。

为什么不写Python呢？因为编程类的专业课有Java，却没有Python，导致我能看懂与维护Java，但是还完全不会Python……虽然都是面向对象。

（这样看来，C++的面向对象就是zhazha——元首）

```
import java.math.BigInteger;
import java.util.HashSet;
import java.util.Scanner;

public class Main
{
    static BigInteger[] a = new BigInteger[100005];
    static boolean[] vis = new boolean[100005];
    int n;
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        for (int i = 1; i < n + 1; i++)
        {
            int temp = sc.nextInt();
            a[i] = BigInteger.valueOf(temp);
        }
        sc.close();
        int cnt;
        BigInteger t;
        HashSet<BigInteger> ans = new HashSet<>();
        for (int i = 1; i < n + 1; i++)
        {
            if (vis[i])
            {
                continue;
            }
            vis[i] = true;
```

```
            t = BigInteger.valueOf(i);
            cnt = 1;
            while (!a[t.intValue()].equals(BigInteger.valueOf(i)))
            {
                vis[a[t.intValue()].intValue()] = true;
                t = a[t.intValue()];
                cnt++;
            }
            ans.add(BigInteger.valueOf(cnt));
        }
        BigInteger ans1 = BigInteger.valueOf(1);
        for (BigInteger e : ans)
        {
            BigInteger gcd = ans1.gcd(e);
            ans1 = ans1.multiply(e.divide(gcd));
        }
        System.out.println(ans1);
    }
}
```

看了标程，用C++补一补题。

```
#include<stdio.h>

#include<map>

using namespace std;

int ww;

int n, p[100010];

map<int,int> fac;

int solve(int pos)
{
    int cnt = 0;
    while(p[pos])
    {
        int nxt = p[pos];
        p[pos] = 0;
        pos = nxt;
        ++cnt;
    }
    return cnt;
}

void factor(int x)
{
```

```
    int i;
    for(i = 2; i*i <= x;++i)
    {
        if( x % i == 0 )
        {
            int cnt = 0;
            while(x % i == 0)
            {
                x /= i;
                ++cnt;
            }
            fac[i] =max(fac[i], cnt);
        }
    }
    if(x != 1)
    {
        fac[x] =max(fac[x], 1);
    }
}

int num[100010],len;

void multiply(int x)
{
    int c = 0;
    int i;
    for(i = 0; i < len; ++i)
    {
        long long tmp = (long long)x * num[i] + c;
        num[i] = int(tmp % 1000000);
        c = int(tmp / 1000000);
    }
    if( c )
    {
        num[len++] = c;
    }
}

int main()
{
    ww = scanf("%d", &n);
    int i;
    for(i = 1; i <= n; ++i)
    {
        ww = scanf("%d", p+i);
    }
    for(i = 1; i <= n; ++i)
    {
        if(p[i])
        {
```

```
            factor(solve(i));
        }
    }
    num[0] = 1, len = 1;
    map<int,int>::iterator it;
    for(it = fac.begin(); it != fac.end(); ++it)
    {
        int now = 1;
        while(it->second--)
        {
            now *= it->first;
        }
        multiply(now);
    }
    for(i = len-1; i >= 0;--i)
    {
        if(i==len-1)
        {
            printf("%d",num[i]);
        }
        else
        {
            printf("%06d",num[i]);
        }
    }
    puts("");
    return 0;
}
```

F

啥都不说，胡佬厉害！打印图形这种足以证明编程水平：

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

int a[105];
int b[105];
int c[105];

int main()
{
    int n, max1 = 0;
    scanf("%d",&n);
    int i;
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
```

```
        max1 =(max1>a[i])?max1:a[i];
    }
    for(i=0;i<n;i++)
    {
        if(a[i] == max1)
        {
            b[i] = 1;
        }
        c[i] = ceil(50.0 * a[i] / max1);
    }
    for(i=0;i<n;i++)
    {
        printf("+");
        int j;
        for(j=0;j<c[i];j++)
        {
            printf("-");
        }
        printf("+\n");
        printf("|");
        for(j=0;j<c[i]-1;j++)
        {
            printf(" ");
        }
        if(b[i] == 1)
        {
            printf("*|%d\n", a[i]);
        }
        else
        {
            (c[i])?printf(" |%d\n", a[i]):printf("|%d\n", a[i]);
        }
        printf("+");
        for(j=0;j<c[i];j++)
        {
            printf("-");
        }
        printf("+\n");
    }
}
```

I

水。

```
#include<stdio.h>

int main()
```

```
{
    printf("0.666667");
    return 0;
}
```

B

以下是补题。

看了标程DFS的思路和我一模一样，但是后面最小生成树没有用Prim而是选择了类似于Kruskal或者其他的优化方法。

```
#include<stdio.h>

#include<algorithm>
#include<vector>

using namespace std;

int fd(int N, int* A, int M, int* B, int K)
{
    if(N==0||M==0)
    {
        return 1<<K;
    }
    if(K==0)
    {
        return 0;
    }
    int X=lower_bound(A, A+N, ((A[0]>>K-1)|1)<<K-1)-A;
    int Y=lower_bound(B, B+M, ((B[0]>>K-1)|1)<<K-1)-B;
    if(X==0 && Y==M || X==N && Y==0)
    {
        return fd(N, A, M, B, K-1) + (1<<(K-1));
    }
    return min(fd(X, A, Y, B, K-1), fd(N-X, A+X, M-Y, B+Y, K-1));
}

long long solve(int N, int* A, int K)
{
    if(N == 1)
    {
        return 0;
    }
    int X =lower_bound(A, A+N, ((A[0] >> (K-1))|1) << (K-1)) - A;
    if(X==0 || X==N)
    {
        return solve(N, A, K-1);
    }
```

```
    return (1ll << (K-1)) + fd(X, A, N-X, A+X, K-1) + solve(X, A,
K-1)+solve(N-X, A+X, K-1);
}

vector<pair<int,int> > G[202020];

int N, A[202020];

bool vis[202020];

void dfs(int x)
{
    vis[x] = 1;
    vector<pair<int,int> >::iterator e;
    for(e=G[x].begin();e!=G[x].end();e++)
    {
        if(!vis[e->first])
        {
            A[e->first] = A[x] ^ e->second;
            dfs(e->first);
        }
    }
}

int main()
{
    scanf("%d", &N);
    int i;
    for(i = 1; i < N; i++)
    {
        int u, v, w;
        scanf("%d%d%d", &u, &v, &w);
        G[u].push_back(make_pair(v,w));
        G[v].push_back(make_pair(u,w));
    }
    A[0] = 0;
    dfs(0);
    sort(A, A+N);
    N =unique(A, A+N) - A;
    printf("%lld\n", solve(N, A, 30));
    return 0;
}
```

C

比赛的时候没有看这道题，后来看了题解，才发现这是一道难得的生成函数的题目。

（好题啊）

题目

W先生正在写序列。如果他写两个长度为K的正整数序列A和B满足：

$$\sum_{i=1}^K a_i=N$$

$$\sum_{i=1}^K b_i=M$$

他会得到:

$$P=\prod_{i=1}^K \min(a_i,b_i)$$

积分。

你想知道他能写的所有可能的序列的总分之和。

输入输出描述

输入第一行包含一个整数T□1≤T≤100□□表示T个测试用例。

接下来的T行每行包含三个整数NMK□1≤N,M≤10^6,1≤K≤min(N,M)□□

输出答案mod 998244353□

做法

生成函数：

$$\sum \min(N,M)x^Ny^M=\frac{xy}{(1-x)(1-y)(1-xy)}$$

枚举(1/(1-xy))^K的次数。

用1/(1-x)^K和1/(1-y)^K补到N-K和M-K□

代码

```
#include<stdio.h>

#define mod 998244353

int fac[2020202],ffac[2020202];

int C(int n,int m)
{
    return 1ll*fac[n]*ffac[m]%mod*ffac[n-m]%mod;
}

int qpow(int n,int k)
{
```

```
    int ret = 1;
    while(k)
    {
        if(k&1)
        {
            ret = 1ll * ret * n % mod;
        }
        n = 1ll * n * n % mod;
        k /= 2;
    }
    return ret;
}

int main()
{
    fac[0] = 1;
    int i;
    for(i = 1; i <= 2e6; i++)
    {
        fac[i] = 1ll * fac[i-1] * i % mod;
    }
    ffac[2000000] = qpow(fac[2000000], mod - 2);
    for(i = 2e6; i >= 1; i--)
    {
        ffac[i - 1] = 1ll * ffac[i] * i % mod;
    }
    int T;
    scanf("%d", &T);
    while(T--)
    {
        int N, M, K;
        scanf("%d%d%d", &N, &M, &K);
        int ans = 0;
        for(i = 0; i <=(N<M?N:M) - K; i++)
        {
            ans += 1ll * C(i + K - 1, K - 1) * C(N - i - 1, K - 1) % mod *
C(M - i - 1, K - 1) % mod;
            if(ans >= mod)
            {
                ans -= mod;
            }
        }
        printf("%d\n",ans);
    }
    return 0;
}
```

J

有趣的计算几何。简单改了改标程，接下来基本可以简化成为C代码了。

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

#define pi acos((long double)-1)
#define eps 1e-11

long double aabs(long double x)
{
    return (long double)(x>0?x:(-x));
}

struct dn
{
    long double x,y;
};

struct dn operator+(struct dn u,struct dn v)
{
    return(struct dn){u.x+v.x,u.y+v.y};
}

struct dn operator-(struct dn u,struct dn v)
{
    return(struct dn){u.x-v.x,u.y-v.y};
}

struct dn operator*(long double u,struct dn v)
{
    return(struct dn){u*v.x,u*v.y};
}

struct dn operator/(struct dn u,long double v)
{
    return(struct dn){u.x/v,u.y/v};
}

long double operator/(struct dn u,struct dn v)
{
    return u.x*v.y-v.x*u.y;
}

long double operator*(struct dn u,struct dn v)
{
    return u.x*v.x+u.y*v.y;
```

```
}

long double dis(struct dn o)
{
    long double temp=o.x*o.x+o.y*o.y>0?o.x*o.x+o.y*o.y:0;
    return sqrt(temp);
}

struct dn dd(long double o)
{
    return(struct dn){cos(o),sin(o)};
}

struct dn g[5];
int top;

void swap(struct dn *a,struct dn *b)
{
    struct dn temp=(*a);
    (*a)=(*b);
    (*b)=temp;
}

void sswap(long double *a,long double *b)
{
    long double temp=(*a);
    (*a)=(*b);
    (*b)=temp;
}

int jd(struct dn o1,long double r1,struct dn o2,long double r2,struct dn
p,struct dn q,struct dn *u,struct dn *v)
{
    top=0;
    struct dn o=(o1-p)*(q-p)*(q-p)/((q-p)*(q-p))+p;
    if(dis(o-o1)<r1-eps)
    {
        g[top]=(o+sqrt(r1*r1-(o1-o)*(o1-o))*(q-p)/dis(q-p));
        top++;
        g[top]=(o-sqrt(r1*r1-(o1-o)*(o1-o))*(q-p)/dis(q-p));
        top++;
    }
    o=(o2-p)*(q-p)*(q-p)/((q-p)*(q-p))+p;
    if(dis(o-o2)<r2-eps)
    {
        g[top]=(o+sqrt(r2*r2-(o2-o)*(o2-o))*(q-p)/dis(q-p));
        top++;
        g[top]=(o-sqrt(r2*r2-(o2-o)*(o2-o))*(q-p)/dis(q-p));
        top++;
    }
```

```
    int i;
    for(i=0;i<top;i++)
    {
        int ok=0;
        int j;
        for(j=0;j<i;j++)
        {
            if(dis(g[i]-g[j])<eps)
            {
                ok=1;
            }
        }
        if(dis(g[i]-o1)>r1+eps||dis(g[i]-o2)>r2+eps)
        {
            ok=1;
        }
        if((g[i]-p)*(q-p)<-eps)
        {
            ok=1;
        }
        if(ok)
        {
            swap(&g[i],&g[top-1]);
            top--;
            i--;
        }
    }
    if(top>0)
    {
        *u=g[0];
    }
    if(top>1)
    {
        *v=g[1];
    }
    if(top==2&&((*u)-p)*(q-p)>((*v)-p)*(q-p))
    {
        swap(&(*u),&(*v));
    }
    return top;
}

long double arc(struct dn o,long double r,struct dn u,struct dn v)
{
    long double w=(v-o)*(u-o)/r/r;
    w=(long double)(w>(-1)?w:(-1));
    w=(long double)(w<1?w:1);
    if((o-u)/(v-u)>0)
    {
        return 0.5*acos(w)*r*r-0.5*aabs((v-o)/(u-o));
    }
```

```
    else
    {
        return pi*r*r-0.5*acos(w)*r*r+0.5*aabs((v-o)/(u-o));
    }
}

long double ar(struct dn o1,long double r1,struct dn o2,long double
r2,struct dn u,struct dn v,int www)
{
    if(dis(o2-o1)<r2-r1+eps)
    {
        if(www)
        {
            return pi*r1*r1;
        }
        else
        {
            return arc(o1,r1,u,v);
        }
    }
    struct dn d1,d2;
    struct dn o=(r1*r1+(o2-o1)*(o2-o1)-r2*r2)/(2*r1*dis(o2-o1))*r1*(o2-
o1)/dis(o2-o1);
    struct dn w=sqrt(r1*r1-o*o)*o/dis(o);
    d1=o1+o+(struct dn){w.y,-w.x};
    d2=o1+o-(struct dn){w.y,-w.x};
    if(o*(o2-o1)<eps)
    {
        swap(&d1,&d2);
    }
    long double aa=arc(o1,r1,d2,d1)+arc(o2,r2,d1,d2);
    if(www)
    {
        return aa;
    }
    if((u-d1)/(d2-d1)<eps&&(v-d1)/(d2-d1)<eps)
    {
        if((0.5*(d1+d2)-u)/(v-u)>eps||(0.5*(d1+d2)-u)/(v-u)>-eps&&(v-
u)*(d2-d1)>0)
        {
            return arc(o2,r2,u,v);
        }
        else
        {
            return aa-arc(o2,r2,v,u);
        }
    }
    if((u-d1)/(d2-d1)>-eps&&(v-d1)/(d2-d1)>-eps)
    {
        if((0.5*(d1+d2)-u)/(v-u)>eps||(0.5*(d1+d2)-u)/(v-u)>-eps&&(v-
```

```
u)*(d1-d2)>0)
        {
            return arc(o1,r1,u,v);
        }
        else
        {
            return aa-arc(o1,r1,v,u);
        }
    }
    if((u-d1)/(d2-d1)>-eps)
    {
        return arc(o1,r1,u,d1)+arc(o2,r2,d1,v)+0.5*aabs((u-d1)/(v-d1));
    }
    else
    {
        return arc(o1,r1,d2,v)+arc(o2,r2,u,d2)+0.5*aabs((u-d2)/(v-d2));
    }
}

long double cc(long double p,long double q,struct dn o1,long double
r1,struct dn o2,long double r2)
{
    if(dis(o1-o2)>r1+r2-eps)
    {
        return 0;
    }
    if(r1>r2)
    {
        swap(&o1,&o2);
        sswap(&r1,&r2);
    }
    if(r1<eps)
    {
        return 0;
    }
    if(dis(o1)<r1-eps&&dis(o2)<r2-eps)
    {
        struct dn u,v;
        jd(o1,r1,o2,r2,(struct dn){0,0},dd(p),&u,&v);
        jd(o1,r1,o2,r2,(struct dn){0,0},dd(q),&v,&u);
        return dis(u-v)<eps?0:ar(o1,r1,o2,r2,v,u,0)+0.5*aabs(u/v);
    }
    else
    {
        struct dn u1,u2,v1,v2;
        int ok1,ok2;
        ok1=jd(o1,r1,o2,r2,(struct dn){0,0},dd(p),&u1,&u2);
        ok2=jd(o1,r1,o2,r2,(struct dn){0,0},dd(q),&v1,&v2);
        if(ok1==1)
        {
            ok1=0;
```

```
        }
        if(ok2==1)
        {
            ok2=0;
        }
        long double aa=ar(o1,r1,o2,r2,(struct dn){0,0},(struct dn){0,0},1);
        if(!ok1&&!ok2)
        {
            struct dn o;
            if(dis(o1-o2)<r2-r1+eps)
            {
                o=o1;
            }
            else
            {
                o=(r1*r1+(o2-o1)*(o2-o1)-r2*r2)/(2*r1*dis(o2-o1))*r1*(o2-
o1)/dis(o2-o1)+o1;
            }
            if(o/dd(p)<-eps&&o/dd(q)>eps)
            {
                return aa;
            }
            else
            {
                return 0;
            }
        }
        if(ok1)
        {
            aa-=ar(o1,r1,o2,r2,u2,u1,0);
        }
        if(ok2)
        {
            aa-=ar(o1,r1,o2,r2,v1,v2,0);
        }
        return aa;
    }
}

int main()
{
    int t=100000;
    scanf("%d",&t);
    int tt;
    for(tt=1;tt<=t;tt++)
    {
        double a,a1,a2,z1,z2,r1,r2;
        a=rand()%89+1;
        a1=rand()%360;
        a2=rand()%360;
```

```
        z1=rand()%2001;
        z2=rand()%2001;
        r1=rand()%2001;
        r2=rand()%2001;
        scanf("%lf%lf%lf%lf%lf%lf%lf",&a,&a1,&a2,&z1,&z2,&r1,&r2);
        a2=a2-a1+360;
        if(a2>=360)
        {
            a2-=360;
        }
        a1=0;
        if(a2>180)
        {
            a2=360-a2;
        }
        a=sin(a/180*pi);
        long double ans=0;
        ans+=cc(0,a2/180*pi*a,z2*dd(a2/180*pi*a),r2,z1*dd(0),r1);
        ans+=cc(0,(180-a2)/180*pi*a,z2*dd(0),r2,z1*dd(-a2/180*pi*a),r1);
        ans+=cc(0,a2/180*pi*a,z2*dd(-(180-a2)/180*pi*a),r2,z1*dd(pi*a),r1);
        ans+=cc(0,(180-a2)/180*pi*a,z2*dd(pi*a),r2,z1*dd((180-
a2)/180*pi*a),r1);
        printf("%.30f\n",(double)ans);
    }
}
```

K

这又是一个大作业题目了。没想到竟然可以简单地用C语言完成。

小型代码管理系统的实现方式