神秘模板库

Toy ASM Truck

Huazhong University of Science and Technology

April 2, 2021

Contents

一切的开始

宏定义

数据结构

ST 表

- 二维

```cpp
int f[maxn][maxn][10][10];
inline int highbit(int x) { return 31 - __builtin_clz(x); }
inline int calc(int x, int y, int xx, int yy, int p, int q) {
    return max(
        max(f[x][y][p][q], f[xx - (1 << p) + 1][yy - (1 << q) + 1][p][q]),
        max(f[xx - (1 << p) + 1][y][p][q], f[x][yy - (1 << q) + 1][p][q])
    );
}
void init() {
    FOR (x, 0, highbit(n) + 1)
    FOR (y, 0, highbit(m) + 1)
        FOR (i, 0, n - (1 << x) + 1)
        FOR (j, 0, m - (1 << y) + 1) {
            if (!x && !y) { f[i][j][x][y] = a[i][j]; continue; }
            f[i][j][x][y] = calc(
                i, j,
                i + (1 << x) - 1, j + (1 << y) - 1,
                max(x - 1, 0), max(y - 1, 0)
            );
        }
}
inline int get_max(int x, int y, int xx, int yy) {
    return calc(x, y, xx, yy, highbit(xx - x + 1), highbit(yy - y + 1));
}
```

数学

模整数类

除法为整除，请乘逆元。

```cpp
template<int P>
struct moint {
    int x;
    moint():x(0){}
    moint(int n){x=n<0?n%P+P:n%P;}
    moint(ll n) {x=n<0?n%P+P:n%P;}
    int get()const{return (int)x;}
    moint &operator+=(moint b){x+=b.x;if(x>=P)x-=P;return *this;}
    moint &operator-=(moint b){x-=b.x;if(x<0)x+=P;return *this;}
    moint &operator*=(moint b){x=1ll*x*b.x%P;return *this;}
    moint &operator/=(moint b){x=x/b.x;return *this;}
    moint &operator%=(moint b){x=x%b.x;return *this;}
    moint operator+(moint b)const{return moint(*this)+=b;}
    moint operator-(moint b)const{return moint(*this)-=b;}
    moint operator*(moint b)const{return moint(*this)*=b;}
    moint operator/(moint b)const{return moint(*this)/=b;}
    moint operator%(moint b)const{return moint(*this)%=b;}
    moint operator+(int b)const{return moint(*this)+=moint(b);}
    moint operator-(int b)const{return moint(*this)-=moint(b);}
    moint operator*(int b)const{return moint(*this)*=moint(b);}
    moint operator/(int b)const{return moint(*this)/=moint(b);}
    moint operator%(int b)const{return moint(*this)%=moint(b);}
    bool operator==(moint b)const{return x==b.x;}
    bool operator>=(moint b)const{return x>=b.x;}
    bool operator!=(moint b)const{return x!=b.x;}
};
typedef moint<998244353> mint;
```

类欧几里得

- $m=\lfloor\frac{an+b}{c}\rfloor$.
- $f(a,b,c,n)=\sum_{i=0}^{n}\lfloor\frac{ai+b}{c}\rfloor$: 当 $a\geq c$ or $b\geq c$ 时，$f(a,b,c,n)=(\frac{a}{c})n(n+1)/2+(\frac{b}{c})(n+1)+f(a \bmod c,b \bmod c,c,n)$; 否则 $f(a,b,c,n)=nm-f(c,c-b-1,a,m-1)$。
- $g(a,b,c,n)=\sum_{i=0}^{n}i\lfloor\frac{ai+b}{c}\rfloor$: 当 $a\geq c$ or $b\geq c$ 时，$g(a,b,c,n)=(\frac{a}{c})n(n+1)(2n+1)/6+(\frac{b}{c})n(n+1)/2+g(a \bmod c,b \bmod c,c,n)$; 否则 $g(a,b,c,n)=\frac{1}{2}(n(n+1)m-f(c,c-b-1,a,m-1)-h(c,c-b-1,a,m-1))$。
- $h(a,b,c,n)=\sum_{i=0}^{n}\lfloor\frac{ai+b}{c}\rfloor^2$: 当 $a\geq c$ or $b\geq c$ 时，$h(a,b,c,n)=(\frac{a}{c})^2n(n+1)(2n+1)/6+(\frac{b}{c})^2(n+1)+(\frac{a}{c})(\frac{b}{c})n(n+1)+h(a \bmod c,b \bmod c,c,n)+2(\frac{a}{c})g(a \bmod c,b \bmod c,c,n)+2(\frac{b}{c})f(a \bmod c,b \bmod c,c,n)$; 否则 $h(a,b,c,n)=nm(m+1)-2g(c,c-b-1,a,m-1)-2f(c,c-b-1,a,m-1)-f(a,b,c,n)$。

```
struct ans{mint f,g,h;};
static ans calc(int a,int b,int c,int n){
    ans ret;
    if(!a){
        ret.f=mint(b/c)*(n+1);
        ret.g=mint(b/c)*n*(n+1)*iv2;
        ret.h=mint(b/c)*(b/c)*(n+1);
        return ret;
    }
    if(a>=c||b>=c){
        ans to=calc(a%c,b%c,c,n);
        ret.f=mint(a/c)*n*(n+1)*iv2+mint(b/c)*(n+1)+to.f;
        ret.g=mint(a/c)*n*(n+1)*(n*2+1)*iv6+mint(b/c)*n*(n+1)*iv2+to.g;
        ret.h=mint(a/c)*(a/c)*n*(n+1)*(n*2+1)*iv6+mint(b/c)*(b/c)*(n+1)+\
            mint(a/c)*(b/c)*n*(n+1)+to.h+mint(a/c)*2*to.g+mint(b/c)*2*to.f;
        return ret;
    }else{
        ll m=(1ll*a*n+b)/c;
        ans to=calc(c,c-b-1,a,m-1);
        ret.f=mint(n*m%P)-to.f;
        ret.g=(mint(m*n%P*(n+1)%P)-to.f-to.h)*iv2;
        ret.h=mint(m*n%P*(m+1)%P)-to.g*2-to.f*2-ret.f;
        return ret;
    }
}
```

Pollard-Rho

```
template<const int test_case>// set 8 usually
struct Pollard_Rho {
    vector<long long> fac;
    long long quick_pow(long long a, long long b, long long mod) {
        long long ans = 1;
        while (b) {
            if (b&1) ans = (__int128)ans*(__int128)a%mod;
            b >>= 1, a = (__int128)a*(__int128)a%mod;
        }
        return ans;
    }
    bool Miller_Rabin(long long n) {// return if n is a prime
        if (n < 3) return n == 2;
        long long a = n-1, b = 0;
        while (a%2 == 0) a /= 2, ++b;
        for (int i = 0, j; i < test_case; i++) {
            long long x = rand()%(n-2)+2, v = quick_pow(x, a, n);
            if (v == 1 || v == n-1) continue;
            for (j = 0; j < b; j++) {
                v = (__int128)v*(__int128)v%n;
                if (v == n-1) break;
            }
            if (j >= b) return false;
        }
        return true;
    }
    long long f(long long x, long long c, long long n) { return ((__int128)x * x + c) % n; }
    long long rho(long long x) {
        long long s = 0, t = 0;
        long long c = (__int128)rand() % (x - 1) + 1;
```

```
        int step = 0, goal = 1;
        long long val = 1;
        for (goal = 1;; goal <<= 1, s = t, val = 1) {
            for (step = 1; step <= goal; ++step) {
                t = f(t, c, x);
                val = (__int128)val * abs(t - s) % x;
                if ((step % 127) == 0) {
                    long long d = __gcd(val, x);
                    if (d > 1) return d;
                }
            }
            long long d = __gcd(val, x);
            if (d > 1) return d;
        }
    }
    void find(long long x) {
        if (x == 1) return;
        if (Miller_Rabin(x)) {
            fac.push_back(x);
            return;
        }
        long long p = x;
        while (p >= x) p = rho(x);
        //while ((x % p) == 0) x /= p;
        find(x/p), find(p);
    }
    vector<long long> factor(long long n) {// return the factors of n
        srand((unsigned)time(NULL));
        fac.clear();
        find(n);
        sort(fac.begin(), fac.end());
        return fac;
    }
};
```

ex-gcd

```
template<typename T>
struct ex_gcd {
    T gcd(const T a, const T b, T &x, T &y) {// x'=x_0+b/gcd, y'=y_0-a/gcd
        if (b == 0) {x = 1, y = 0; return a; }
        T d = gcd(b, a%b, x, y);
        T t = x;
        x = y;
        y = t - a/b*y;
        return d;
    }
    T inv(const T a, const T m) {// return -1 if inv is not exist
        if (a == 0 || m <= 1) return -1;
        T x, y, d = gcd(a, m, x, y);
        if (d != 1) return -1;
        return (x%m+m)%m;
    }
} ;
```

crt

```
template<typename T>
struct crt {
    ex_gcd<T> *exgcd = new ex_gcd<T>();
    T cal(const T *a, const T *m, const int n) {// a[1..n], m[1..n], gcd(m_i) = 1
        T M = 1, ans = 0;
        for (int i = 1; i <= n; i++) M *= m[i];
        for (int i = 1; i <= n; i++)
            (ans += (__int128)a[i]*(M/m[i])%M*exgcd->inv(M/m[i], m[i])%M) %= M;
        return ans;
    }
} ;
```

ex-crt

```cpp
template<typename T>
struct ex_crt {
    ex_gcd<T> *exgcd = new ex_gcd<T>();
    T cal(T *a, T *m, const int n) {// a[1..n], m[1..n], return -1 if no ans
        T x, y, gcd, lcm;
        for (int i = 2; i <= n; i++) {
            gcd = exgcd->gcd(m[1], m[i], x, y);
            if ((a[i]-a[1])%gcd) return -1;
            lcm = (__int128)m[1]*m[i]/gcd;
            x = (__int128)x*(a[i]-a[1])/gcd%lcm;
            gcd = m[i]/gcd;
            x = (x%gcd+gcd)%gcd;
            a[1] = ((__int128)m[1]*x%lcm+a[1])%lcm, m[1] = lcm;
        }
        return a[1];
    }
} ;
```

Meissel-Lehmer

求解 1e11 内的质数个数，约为 $O(n^{2/3})$。

```cpp
namespace pcf{
#define chkbit(ar, i) (((ar[(i) >> 6]) & (1 << (((i) >> 1) & 31))))
#define setbit(ar, i) (((ar[(i) >> 6]) |= (1 << (((i) >> 1) & 31))))
#define isprime(x) (( (x) && ((x)&1) && (!chkbit(ar, (x)))) || ((x) == 2))
    const int MAXN=100;
    const int MAXM=10001;
    const int MAXP=40000;
    const int MAX=400000;
    long long dp[MAXN][MAXM];
    unsigned int ar[(MAX >> 6) + 5] = {0};
    int len = 0, primes[MAXP], counter[MAX];
    void Sieve(){
        setbit(ar, 0), setbit(ar, 1);
        for (int i = 3; (i * i) < MAX; i++, i++){
            if (!chkbit(ar, i)){
                int k = i << 1;
                for (int j = (i * i); j < MAX; j += k) setbit(ar, j);
            }
        }
        for (int i = 1; i < MAX; i++){
            counter[i] = counter[i - 1];
            if (isprime(i)) primes[len++] = i, counter[i]++;
        }
    }
    void init(){
        Sieve();
        for (int n = 0; n < MAXN; n++){
            for (int m = 0; m < MAXM; m++){
                if (!n) dp[n][m] = m;
                else dp[n][m] = dp[n - 1][m] - dp[n - 1][m / primes[n - 1]];
            }
        }
    }
    long long phi(long long m, int n){
        if (n == 0) return m;
        if (primes[n - 1] >= m) return 1;
        if (m < MAXM && n < MAXN) return dp[n][m];
        return phi(m, n - 1) - phi(m / primes[n - 1], n - 1);
    }
    long long Lehmer(long long m){
        if (m < MAX) return counter[m];
        long long w, res = 0;
        int i, a, s, c, x, y;
        s = sqrt(0.9 + m), y = c = cbrt(0.9 + m);
        a = counter[y], res = phi(m, a) + a - 1;
        for (i = a; primes[i] <= s; i++) res = res - Lehmer(m / primes[i]) + Lehmer(primes[i]) - 1;
        return res;
```

```
    }
}
int main(){
    pcf::init();
    long long n;
    while (scanf("%lld", &n) != EOF){
        printf("%lld\n",pcf::Lehmer(n));
    }
    return 0;
}
```

Cipolla 二次剩余

- $x^2 \equiv n (\mathrm{mod} P)$
- 仅有两个解，返回小的那个，另一个是相反数。
- 大概 1s 能跑 1e5 个数

```
inline int qpow(int a,int b){
    int q=1;while(b){if(b&1)q=1ll*q*a%P;a=1ll*a*a%P;b>>=1;}return q;
}
namespace Cipolla{
    ll w,a;
    struct node{
        ll x,y;
        node friend operator *(node x,node y){
            node z;
            z.x=(x.x*y.x%P+x.y*y.y%P*w%P)%P;
            z.y=(x.x*y.y%P+x.y*y.x%P)%P;
            return z;
        }
    }u,v;
    inline node Cqpow(node a,ll b){
        node q;q.x=1;q.y=0;
        while(b){if(b&1) q=q*a;a=a*a;b>>=1;}
        return q;
    }
    inline ll cipolla(int n){
        n%=P;srand(0x20010412);
        if(P==2) return n;
        if(!n) return n;
        if(qpow(n,(P-1)/2)==P-1) return -1;
        while(1){
            a=rand()%P;
            w=(a*a-n+P)%P;
            if(qpow(w%P,(P-1)/2)==P-1) break;
        }
        u.x=a,u.y=1;
        u=Cqpow(u,(P+1)/2);
        ll fir=u.x,sec=P-u.x;
        if(fir>sec)swap(fir,sec);
        return fir;
    }
}
```

BSGS

```
template<typename T>
struct BSGS {
    T cal(T a, T b, T c) { // return a^x = b (mod c), gcd(a, c) = 1
        mp.clear();
        T tim = ceil(sqrt(c)), tmp = b%c;
        for (int i = 0; i <= tim; i++) {
            mp[tmp] = i; tmp = (__int128)tmp*a%c;
        }
        T t = tmp = quick_pow(a, tim, c);
        for (int i = 1; i <= tim; i++) {
            if (mp.count(tmp)) return tim*i-mp[tmp];
            tmp = (__int128)tmp*t%c;
        }
        return -1;
```

```
    }
} ;
```

exBSGS

```
template<typename T>
struct exBSGS {
    T cal(T a, T b, T c) { // return a^x = b (mod c)
        if (b == 1) return 0;
        T cnt = 0, d = 1, t;
        while ((t = __gcd(a, c)) != 1) {
            if (b%t) return -1;
            ++cnt, b /= t, c /= t, d = (__int128)d*(a/t)%c;
            if (d == b) return cnt;
        }
        mp.clear();
        T tim = ceil(sqrt(c)), tmp = b%c;
        for (int i = 0; i <= tim; i++) {
            mp[tmp] = i; tmp = (__int128)tmp*a%c;
        }
        t = tmp = quick_pow(a, tim, c); tmp = (__int128)tmp*d%c;
        for (int i = 1; i <= tim; i++) {
            if (mp.count(tmp)) return tim*i-mp[tmp]+cnt;
            tmp = (__int128)tmp*t%c;
        }
        return -1;
    }
} ;
```

```
template<typename T>
struct exLucas {
    T quick_pow(T a, T b, T p) {
        T ans = 1;
        while (b) {
            if (b&1) ans = (__int128)ans*a%p;
            b >>= 1, a = (__int128)a*a%p;
        }
        return ans;
    }
    void ex_gcd(T a, T b, T &x, T &y) {
        if (b == 0) {x = 1, y = 0; return; }
        ex_gcd(b, a%b, x, y);
        T t = x; x = y, y = t-a/b*y;
    }
    T inv(T a, T p) {
        T x, y; ex_gcd(a, p, x, y);
        return (x%p+p)%p;
    }
    T mul(T n, T pi, T pk) {
        if (!n) return 1;
        T ans = 1;
        for (int i = 2; i <= pk; i++) if (i%pi != 0) ans = (__int128)ans*i%pk;
        ans = quick_pow(ans, n/pk, pk);
        for (int i = 2; i <= n%pk; i++) if (i%pi != 0) ans = (__int128)ans*i%pk;
        return (__int128)ans*mul(n/pi, pi, pk)%pk;
    }
    T C(T n, T m, T pi, T pk, T p) {
        T a = mul(n, pi, pk), b = mul(m, pi, pk), c = mul(n-m, pi, pk);
        T k = 0;
        for (T i = n; i; i /= pi) k += i/pi;
        for (T i = m; i; i /= pi) k -= i/pi;
        for (T i = n-m; i; i /= pi) k -= i/pi;
        return (__int128)a*inv(b, pk)%pk*inv(c, pk)%pk*quick_pow(pi, k, pk)%pk;
    }
    T ex_lucas(T n, T m, T p) {
        T ans = 0;
        for (T i = 2, x = p; i <= x; i++)
            if (x%i == 0) {
                T k = 1; while (x%i == 0) k *= i, x /= i;
                (ans += (__int128)C(n, m, i, k, p)*(p/k)%p*inv(p/k, k)%p) %= p;
            }
```

```
        return ans;
    }
} ;
```

图论

LCA

- 倍增

```
void dfs(int u, int fa) {
    pa[u][0] = fa; dep[u] = dep[fa] + 1;
    FOR (i, 1, SP) pa[u][i] = pa[pa[u][i - 1]][i - 1];
    for (int& v: G[u]) {
        if (v == fa) continue;
        dfs(v, u);
    }
}


int lca(int u, int v) {
    if (dep[u] < dep[v]) swap(u, v);
    int t = dep[u] - dep[v];
    FOR (i, 0, SP) if (t & (1 << i)) u = pa[u][i];
    FORD (i, SP - 1, -1) {
        int uu = pa[u][i], vv = pa[v][i];
        if (uu != vv) { u = uu; v = vv; }
    }
    return u == v ? u : pa[u][0];
}
```

计算几何

二维几何：点与向量

```
#define y1 yy1
#define nxt(i) ((i + 1) % s.size())
typedef double LD;
const LD PI = 3.14159265358979323846;
const LD eps = 1E-10;
int sgn(LD x) { return fabs(x) < eps ? 0 : (x > 0 ? 1 : -1); }
struct L;
struct P;
typedef P V;
struct P {
    LD x, y;
    explicit P(LD x = 0, LD y = 0): x(x), y(y) {}
    explicit P(const L& l);
};
struct L {
    P s, t;
    L() {}
    L(P s, P t): s(s), t(t) {}
};

P operator + (const P& a, const P& b) { return P(a.x + b.x, a.y + b.y); }
P operator - (const P& a, const P& b) { return P(a.x - b.x, a.y - b.y); }
P operator * (const P& a, LD k) { return P(a.x * k, a.y * k); }
P operator / (const P& a, LD k) { return P(a.x / k, a.y / k); }
inline bool operator < (const P& a, const P& b) {
    return sgn(a.x - b.x) < 0 || (sgn(a.x - b.x) == 0 && sgn(a.y - b.y) < 0);
}
bool operator == (const P& a, const P& b) { return !sgn(a.x - b.x) && !sgn(a.y - b.y); }
P::P(const L& l) { *this = l.t - l.s; }
ostream &operator << (ostream &os, const P &p) {
    return (os << "(" << p.x << "," << p.y << ")");
}
istream &operator >> (istream &is, P &p) {
```

```
    return (is >> p.x >> p.y);
}

LD dist(const P& p) { return sqrt(p.x * p.x + p.y * p.y); }
LD dot(const V& a, const V& b) { return a.x * b.x + a.y * b.y; }
LD det(const V& a, const V& b) { return a.x * b.y - a.y * b.x; }
LD cross(const P& s, const P& t, const P& o = P()) { return det(s - o, t - o); }
// --------------------------------------------
```

完整的板板 by zcs

```
#include <bits/stdc++.h>
using namespace std;
const double pi=acos(-1.0);   //高精度圆周率
const double eps=1e-8;        //偏差值
const int maxp=200005;          //点的数量
int sgn(double x)             //判断 x 是否为 0
{
    if(fabs(x)<eps) return 0;
    else return x<0?-1:1;     //小于 0 返回-1，大于 0 返回 1
}
int Dcmp(double x,double y)   //比较浮点数大小
{
    if(fabs(x-y)<eps) return 0;
    else return x<y?-1:1;
}
//----------------------------------平面几何：点和线-------------------------------------
struct Point
{
    double x,y;
    Point(){}
    Point(double x,double y):x(x),y(y){}
    Point operator + (Point B){return Point(x+B.x,y+B.y);}
    Point operator - (Point B){return Point(x-B.x,y-B.y);}
    Point operator * (double k){return Point(x*k,y*k);}                 //长度扩大 k 倍
    Point operator / (double k){return Point(x/k,y/k);}                 //长度缩小 k 倍
    bool operator == (Point B){return sgn(x-B.x)==0 && sgn(y-B.y)==0;}
};
typedef Point Vector;
double Dot(Vector A,Vector B){return A.x*B.x+A.y*B.y;}                  //向量点乘
double Len(Vector A){return sqrt(Dot(A,A));}                            //向量取模
double Angle(Vector A,Vector B){return acos(Dot(A,B)/Len(A)/Len(B));}   //向量 A 和 B 夹角
double Cross(Vector A,Vector B){return A.x*B.y-A.y*B.x;}                //向量叉乘
//double Area(Point A,Point B,Point C){return Cross(B-A,C-A);}          //三角形面积 2 倍
double Distance(Point A,Point B){return hypot(A.x-B.x,A.y-B.y);}        //两点距离
Vector Normal(Vector A){return Vector(-A.y/Len(A),A.x/Len(A));}         //向量 A 的单位 * 法 * 向量
bool Parallel(Vector A,Vector B){return sgn(Cross(A,B))==0;}            //平行
Vector Rotate(Vector A,double rad)                                      //向量旋转
{return Vector(A.x*cos(rad)-A.y*sin(rad),A.x*sin(rad)+A.y*cos(rad));}
struct Line
{
    Point p1,p2;
    Line(){}
    Line(Point p1,Point p2):p1(p1),p2(p2){}                             //两点确定直线
    Line(Point p,double angle)                                          //点 + 倾斜角
    {
        p1=p;
        if(sgn(angle-pi/2)==0){p2=(p1+Point(0,1));}
        else {p2=(p1+Point(1,tan(angle)));}
    }
    Line(double a,double b,double c)                                    //ax+by+c=0;
    {
        if(sgn(a)==0) {p1=Point(0,-c/b);p2=Point(1,-c/b);}
        else if(sgn(b)==0) {p1=Point(-c/a,0);p2=Point(-c/a,1);}
        else {p1=Point(0,-c/b);p2=Point(1,(-c-a)/b);}
    }
};
typedef Line Segment;
//直线倾斜角，返回值 [0,pi);
double Line_angle(Line v)
{
```

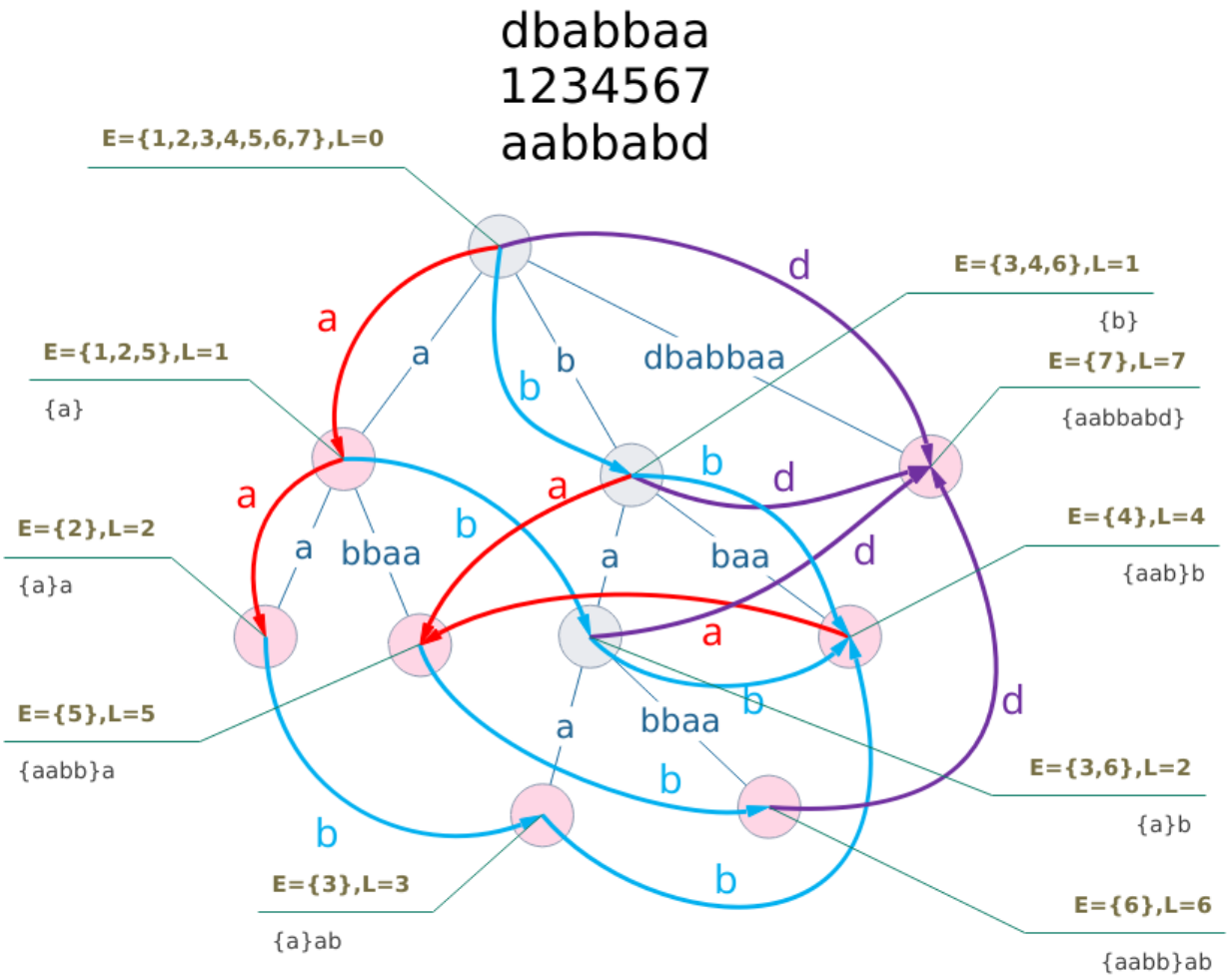
```
    double k=atan2(v.p2.y-v.p1.y,v.p2.x-v.p1.x);
    if(sgn(k)<0) k+=pi;
    if(sgn(k-pi)==0) k-=pi;
        return k;
}
//点和直线关系
int Point_line_relation(Point p,Line v)
{
    int c=sgn(Cross(p-v.p1,v.p2-v.p1));
    if(c<0) return 1;                           //1:p 在 v 左侧
    if(c>0) return 2;                           //2:p 在 v 右侧
    return 0;                                   //0:p 在 v 上
}
//点和线段关系: 0 为 p 不在线段 v 上; 1 为 p 在线段 v 上
bool Point_on_seg(Point p,Segment v)
{
    return sgn(Cross(p-v.p1,v.p2-v.p1))==0 && sgn(Dot(p-v.p1,p-v.p2))<=0;
}
//两直线的关系: 0 为平行, 1 为重合, 2 为相交
int Line_relation(Line v1,Line v2)
{
    if(sgn(Cross(v1.p2-v1.p1,v2.p2-v2.p1))==0)
    {
        if(Point_line_relation(v1.p1,v2)==0) return 1;
        else return 0;
    }
    return 2;
}
//点到直线距离
double Dis_point_line(Point p,Line v)
{
    return fabs(Cross(p-v.p1,v.p2-v.p1))/Distance(v.p1,v.p2);
}
//点在直线上的投影
Point Point_line_proj(Point p,Line v)
{
    double k=Dot(v.p2-v.p1,p-v.p1)/Dot(v.p2-v.p1,v.p2-v.p1);
    return v.p1+(v.p2-v.p1)*k;
}
//点 p 对直线 v 的对称点
Point Point_line_symmetry(Point p,Line v)
{
    Point q=Point_line_proj(p,v);
    return Point(2*q.x-p.x,2*q.y-p.y);
}
//点到线段的距离
double Dis_point_seg(Point p,Segment v)
{
    if(sgn(Dot(p-v.p1,v.p2-v.p1))<0 || sgn(Dot(p-v.p2,v.p1-v.p2))<0)
        return min(Distance(p,v.p1),Distance(p,v.p2));                  //点的投影不在线段上
    return Dis_point_line(p,v);                                         //点的投影在线段上
}
//求两直线 ab 和 cd 的交点, 在调用前要保证两直线不平行或重合
Point Cross_point(Point a,Point b,Point c,Point d)
{
    double s1=Cross(b-a,c-a);
    double s2=Cross(b-a,d-a);
    return Point(c.x*s2-d.x*s1,c.y*s2-d.y*s1)/(s2-s1);
}
//线段 ab 和 cd 是否相交
bool Cross_segment(Point a,Point b,Point c,Point d)
{
    double c1=Cross(b-a,c-a),c2=Cross(b-a,d-a);
    double d1=Cross(d-c,a-c),d2=Cross(d-c,b-c);
    return sgn(c1)*sgn(c2)<=0 && sgn(d1)*sgn(d2)<=0;
}
//----------------------------------平面几何: 多边形-----------------------------------
struct Polygon
{
    int n;
    Point p[maxp];                      //从 0 开始
```

```
    Line v[maxp];
};
//极角排序
bool Polar_angle_cmp(Point a,Point b)
{
    if(Cross(a,b)==0) return a.x<b.x;
    else return Cross(a,b)>0;
}
//按照 x 大小排序（计算凸包使用）
bool Hull_cmp(Point A,Point B)
{
    return sgn(A.x-B.x)<0 || (sgn(A.x-B.x)==0 && sgn(A.y-B.y)<0);
}
//判断点和任意多边形的关系：3 为点上；2 为边上；1 为内部；0 为外部
int Point_in_polygon(Point pt,Point *p,int n)                    //点 pt，多边形 *p
{
    for(int i=0;i<n;i++)
        if(p[i]==pt) return 3;
    for(int i=0;i<n;i++)
    {
        Line v=Line(p[i],p[(i+1)%n]);
        if(Point_on_seg(pt,v)) return 2;
    }
    int num=0;
    for(int i=0;i<n;i++)
    {
        int j=(i+1)%n;
        int c=sgn(Cross(pt-p[j],p[i]-p[j]));
        int u=sgn(p[i].y-pt.y);
        int v=sgn(p[j].y-pt.y);
        if(c>0 && u<0 && v>=0) num++;
        if(c<0 && u>=0 && v<0) num--;
    }
    return num!=0;
}
//多边形面积
double Polygon_area(Point *p,int n)
{
    double area=0;
    for(int i=0;i<n;i++)
        area+=Cross(p[i],p[(i+1)%n]);
    return area/2;
}
//求多边形重心
Point Polygon_center(Point *p,int n)
{
    Point ans(0,0);
    if(Polygon_area(p,n)==0) return ans;
    for(int i=0;i<n;i++)
        ans=ans+(p[i]+p[(i+1)%n])*Cross(p[i],p[(i+1)%n]);
    return ans/Polygon_area(p,n)/6;
}
//Convex_hull() 求凸包，凸包顶点放在 ch 中，返回值是凸包的顶点数
int Convex_hull(Point *p,int n,Point *ch)
{
    sort(p,p+n,Hull_cmp);
    n=unique(p,p+n)-p;
    int v=0;
    //求下凸包，如果 p[i] 是右拐的，则不在凸包上，往回退
    for(int i=0;i<n;i++)
    {
        while(v>1 && sgn(Cross(ch[v-1]-ch[v-2],p[i]-ch[v-2]))<=0)
            v--;
        ch[v++]=p[i];
    }
    int j=v;
    //求上凸包
    for(int i=n-2;i>=0;i--)
    {
        while(v>j && sgn(Cross(ch[v-1]-ch[v-2],p[i]-ch[v-2]))<=0)
            v--;
```

```
        ch[v++]=p[i];
    }
    if(n>1) v--;
    return v;
}
//----------------------------------平面几何：圆-----------------------------------
struct Circle
{
    Point c;                            //圆心
    double r;
    Circle(){}
    Circle(Point c,double r):c(c),r(r){}
    Circle(double x,double y,double _r){c=Point(x,y);r=_r;}
};
//点和圆的关系：0 为圆内，1 为圆上，2 为圆外
int Point_circle_relation(Point p,Circle C)
{
    double dst=Distance(p,C.c);
    if(sgn(dst-C.r)<0) return 0;
    if(sgn(dst-C.r)==0) return 1;
    return 2;
}
//直线和圆的关系：0 为直线和圆相交，1 为直线和圆相切，2 为直线和圆相离
int Line_circle_relation(Line v,Circle C)
{
    double dst=Dis_point_line(C.c,v);
    if(sgn(dst-C.r)<0) return 0;
    if(sgn(dst-C.r)==0) return 1;
    return 2;
}
//线段和圆的关系：0 为线段和圆相交，1 为线段和圆相切，2 为线段和圆相离
int Seg_circle_relation(Segment v,Circle C)
{
    double dst=Dis_point_seg(C.c,v);
    if(sgn(dst-C.r)<0) return 0;
    if(sgn(dst-C.r)==0) return 1;
    return 2;
}
//直线和圆的交点，pa,pb 是交点，返回值是交点个数
int Line_cross_circle(Line v,Circle C,Point &pa,Point &pb)
{
    if(Line_circle_relation(v,C)==2) return 0;          //无交点
    Point q=Point_line_proj(C.c,v);
    double d=Dis_point_line(C.c,v);
    double k=sqrt(C.r*C.r-d*d);
    if(sgn(k)==0)
    {
        pa=q;pb=q;return 1;
    }
    Point n=(v.p2-v.p1)/Len(v.p2-v.p1);                 //直线的单位向量
    pa=q+n*k;
    pb=q-n*k;
    return 2;
}
```

字符串

后缀自动机



多项式

NTT 模数

```
NTTPrimes = {1053818881, 1051721729, 1045430273, 1012924417, 1007681537, 1004535809, 998244353, 985661441,
 ↪  976224257, 975175681};
NTTPrimitiveRoots = {7, 6, 3, 5, 3, 3, 3, 3, 3, 17};
```

FFT

```
namespace FFT{
    const db pi=acos(-1);
    struct cp{
        db re,im;
        cp(db _re=0,db _im=0){re=_re;im=_im;}
        cp operator +(cp b){return cp(re+b.re,im+b.im);}
        cp operator -(cp b){return cp(re-b.re,im-b.im);}
        cp operator *(cp b){return cp(re*b.re-im*b.im,re*b.im+im*b.re);}
    };
    int r[N];cp c[N<<1];
    inline void fft(cp *a,int f,int n){
        rep(i,0,n-1) if(r[i]>i) swap(a[r[i]],a[i]);
        for(int i=1;i<n;i<<=1){
```

```
            cp wn(cos(pi/i),f*sin(pi/i));
            for(int j=0,p=(i<<1);j<n;j+=p){
                cp w(1,0);
                for(int k=0;k<i;++k,w=w*wn){
                    cp x=a[j+k],y=w*a[j+k+i];
                    a[j+k]=x+y;a[j+k+i]=x-y;
                }
            }
        }
        if(f==-1){rep(i,0,n-1) a[i].re/=n,a[i].im/=n;}
    }
    inline int mul(db *a,db *b,int n,int m){
        n+=m;rep(i,0,n) c[i]=cp(a[i],b[i]);
        int l=0;m=n;for(n=1;n<=m;n<<=1) ++l;
        rep(i,0,n-1) r[i]=(r[i>>1]>>1)|((i&1)<<(l-1));
        rep(i,m+1,n) c[i]=cp(0,0);
        fft(c,1,n);rep(i,0,n-1) c[i]=c[i]*c[i];
        fft(c,-1,n);
        rep(i,0,m) a[i]=c[i].im/2;
        return n;
    }
}
```

NTT

```
namespace NTT{
    const int P=998244353,g=3,ig=332748118;
    inline int qpow(int a,int b){int q=1;while(b){if(b&1)q=1LL*q*a%P;a=1LL*a*a%P;b>>=1;}return q;}
    int r[N],ow[N],inv[N];
    inline void ntt(int *a,int f,int n){
        rep(i,0,n-1) if(r[i]>i) swap(a[i],a[r[i]]);
        for(int i=1;i<n;i<<=1){
            int wn=qpow(f,(P-1)/(i<<1));
            ow[0]=1;rep(k,1,i-1) ow[k]=1LL*ow[k-1]*wn%P;
            for(int j=0,p=(i<<1);j<n;j+=p){
                for(int k=0;k<i;++k){
                    int x=a[j+k],y=1LL*ow[k]*a[j+k+i]%P;
                    a[j+k]=(x+y)%P;a[j+k+i]=(x+P-y)%P;
                }
            }
        }
        if(f==ig){
            int iv=qpow(n,P-2);
            rep(i,0,n-1) a[i]=1LL*a[i]*iv%P;
        }
    }
    int tma[N],tmb[N];
    inline int mul(int *a,int *b,int n,int m,int ci){
        int _n=n,_m=m,l=0;m+=n;for(n=1;n<=m;n<<=1) ++l;
        rep(i,0,n-1) r[i]=(r[i>>1]>>1)|((i&1)<<(l-1));
        rep(i,0,n-1) tma[i]=a[i];rep(i,0,n-1) tmb[i]=b[i];
        rep(i,_n+1,n) tma[i]=0;rep(i,_m+1,n) tmb[i]=0;
        ntt(tma,g,n);ntt(tmb,g,n);
        while(ci){
            if(ci&1) rep(i,0,n-1) tma[i]=1LL*tma[i]*tmb[i]%P;
            rep(i,0,n-1) tmb[i]=1LL*tmb[i]*tmb[i]%P;
            ci>>=1;
        }
        ntt(tma,ig,n);
        rep(i,0,n-1) a[i]=tma[i];
        return n;
    }
}
inline void prepare(){
    //NTT inv
    using NTT::inv;using NTT::P;
    inv[1]=1;rep(i,2,N-1) inv[i]=1LL*(P-P/i)*inv[P%i]%P;
}
```

完整的板板 by hls

```
/*
    NTTPrimes = {1053818881, 1051721729, 1045430273, 1012924417, 1007681537, 1004535809, 998244353, 985661441,
    976224257, 975175681};
    NTTPrimitiveRoots = {7, 6, 3, 5, 3, 3, 3, 3, 3, 17};
*/
//poly start
namespace Poly{
    const int N=6e5+10;
    namespace NTT{
        const int P=998244353,g=3,ig=332748118;
        inline int qpow(int a,int b){int q=1;while(b){if(b&1)q=1LL*q*a%P;a=1LL*a*a%P;b>>=1;}return q;}
        int r[N],ow[N],inv[N];
        inline void ntt(int *a,int f,int n){
            rep(i,0,n-1) if(r[i]>i) swap(a[i],a[r[i]]);
            for(int i=1;i<n;i<<=1){
                int wn=qpow(f,(P-1)/(i<<1));
                ow[0]=1;rep(k,1,i-1) ow[k]=1LL*ow[k-1]*wn%P;
                for(int j=0,p=(i<<1);j<n;j+=p){
                    for(int k=0;k<i;++k){
                        int x=a[j+k],y=1LL*ow[k]*a[j+k+i]%P;
                        a[j+k]=(x+y)%P;a[j+k+i]=(x+P-y)%P;
                    }
                }
            }
            if(f==ig){
                int iv=qpow(n,P-2);
                rep(i,0,n-1) a[i]=1LL*a[i]*iv%P;
            }
        }
        int tma[N],tmb[N];
        inline int mul(int *a,int *b,int n,int m,int ci){
            int _n=n,_m=m,l=0;m+=n;for(n=1;n<=m;n<<=1) ++l;
            rep(i,0,n-1) r[i]=(r[i>>1]>>1)|((i&1)<<(l-1));
            rep(i,0,n-1) tma[i]=a[i];rep(i,0,n-1) tmb[i]=b[i];
            rep(i,_n+1,n) tma[i]=0;rep(i,_m+1,n) tmb[i]=0;
            ntt(tma,g,n);ntt(tmb,g,n);
            while(ci){
                if(ci&1) rep(i,0,n-1) tma[i]=1LL*tma[i]*tmb[i]%P;
                rep(i,0,n-1) tmb[i]=1LL*tmb[i]*tmb[i]%P;
                ci>>=1;
            }
            ntt(tma,ig,n);
            rep(i,0,n-1) a[i]=tma[i];
            return n;
        }
    }
    namespace FFT{
        const db pi=acos(-1);
        struct cp{
            db re,im;
            cp(db _re=0,db _im=0){re=_re;im=_im;}
            cp operator +(cp b){return cp(re+b.re,im+b.im);}
            cp operator -(cp b){return cp(re-b.re,im-b.im);}
            cp operator *(cp b){return cp(re*b.re-im*b.im,re*b.im+im*b.re);}
        };
        int r[N];cp c[N<<1];
        inline void fft(cp *a,int f,int n){
            rep(i,0,n-1) if(r[i]>i) swap(a[r[i]],a[i]);
            for(int i=1;i<n;i<<=1){
                cp wn(cos(pi/i),f*sin(pi/i));
                for(int j=0,p=(i<<1);j<n;j+=p){
                    cp w(1,0);
                    for(int k=0;k<i;++k,w=w*wn){
                        cp x=a[j+k],y=w*a[j+k+i];
                        a[j+k]=x+y;a[j+k+i]=x-y;
                    }
                }
            }
            if(f==-1){rep(i,0,n-1) a[i].re/=n,a[i].im/=n;}
```

```
        }
        inline int mul(db *a,db *b,int n,int m){
            n+=m;rep(i,0,n) c[i]=cp(a[i],b[i]);
            int l=0;m=n;for(n=1;n<=m;n<<=1) ++l;
            rep(i,0,n-1) r[i]=(r[i>>1]>>1)|((i&1)<<(l-1));
            rep(i,m+1,n) c[i]=cp(0,0);
            fft(c,1,n);rep(i,0,n-1) c[i]=c[i]*c[i];
            fft(c,-1,n);
            rep(i,0,m) a[i]=c[i].im/2;
            return n;
        }
    }
    using namespace NTT;
    ll w,a;
    struct node{
        ll x,y;
        node friend operator *(node x,node y){
            node z;
            z.x=(x.x*y.x%P+x.y*y.y%P*w%P)%P;
            z.y=(x.x*y.y%P+x.y*y.x%P)%P;
            return z;
        }
    }u,v;
    inline node Cqpow(node a,ll b){
        node q;q.x=1;q.y=0;
        while(b){if(b&1) q=q*a;a=a*a;b>>=1;}
        return q;
    }
    inline ll cipolla(int n,int P){
        n%=P;srand(0x20010412);
        if(P==2) return 1;
        if(qpow(n,(P-1)/2)==P-1) return -1;
        while(1){
            a=rand()%P;
            w=(a*a-n+P)%P;
            if(qpow(w%P,(P-1)/2)==P-1) break;
        }
        u.x=a,u.y=1;
        u=Cqpow(u,(P+1)/2);
        ll fir=u.x,sec=P-u.x;
        if(fir>sec)swap(fir,sec);
        return fir;
    }
    inline void derivative(int *a,int *b,int n){
        rpe(i,n-2,0) b[i]=1LL*a[i+1]*(i+1)%P;
        b[n-1]=0;
    }
    inline void integral(int *a,int *b,int n){
        rpe(i,n-1,1) b[i]=1LL*a[i-1]*inv[i]%P;
        b[0]=0;
    }
    inline void differential(int *a,int *b,int n){
        rep(i,1,n-1) b[i]=(a[i]+P-a[i-1])%P;
        b[0]=0;
    }
    int tf[N],tg[N];
    inline void inverse(int *f,int *g,int n){
        if(n==1){
            g[0]=qpow(f[0],P-2);return;
        }
        inverse(f,g,n>>1);
        rep(i,0,n-1) tf[i]=f[i],tg[i]=g[i];
        int tmp=mul(tf,tg,n-1,n-1,2);
        rep(i,0,n-1) g[i]=((-tf[i]+2LL*g[i])%P+P)%P;
        rep(i,0,tmp) tf[i]=tg[i]=0;
    }
    int ta[N],tb[N];
    inline void sqrt(int *a,int *b,int n){
        if(n==1){
            b[0]=cipolla(a[0],P);//debug(b[0]);debug(1LL*b[0]*b[0]%P);
            return;
```

```
        }
        sqrt(a,b,n>>1);rep(i,n,(n<<1)) b[i]=0;
        inverse(b,tb,n);
        rep(i,0,n-1) ta[i]=a[i];
        mul(ta,tb,n-1,n-1,1);//debug(ta[0]);debug(b[0]);
        rep(i,0,n-1) b[i]=1LL*(b[i]+ta[i])%P*inv[2]%P;
        rep(i,0,n<<1) ta[i]=tb[i]=0;
    }
    inline void ln(int *a,int *b,int n){
        inverse(a,ta,n);
        derivative(a,b,n);
        mul(b,ta,n,n,1);
        integral(b,b,n);
    }
    inline void exp(int *a,int *b,int n){
        if(n==1){
            b[0]=1;return;
        }
        exp(a,b,n>>1);
        ln(b,tb,n);rep(i,0,n-1) ta[i]=a[i];++ta[0];
        rep(i,0,n-1) ta[i]-=tb[i];
        mul(ta,b,n,n,1);rep(i,0,n-1) b[i]=ta[i];
        rep(i,0,n<<1) ta[i]=tb[i]=0;
    }
}
//
using namespace Poly;
inline void prepare(){
    //NTT
    using NTT::inv;using NTT::P;
    inv[1]=1;rep(i,2,N-1) inv[i]=1LL*(P-P/i)*inv[P%i]%P;
}
int n,A[N],B[N];
int main(){
    prepare();
    n=read();rep(i,0,n) A[i]=read();
    int len=1;for(;len<=n;len<<=1);
/*
    sqrt(A,B,len);
    rep(i,0,n) printf("%d ",B[i]);pts;
//    debug(1LL*B[0]*B[0]%P);
    NTT::mul(B,B,n,n,1);
    rep(i,0,n) printf("%d ",B[i]);pts;
*/ //sqrt 1e5 uoj 333ms
/*
    ln(A,B,len);
    rep(i,0,n) printf("%d ",B[i]);pts;
    exp(B,A,len);
    rep(i,0,n) printf("%d ",A[i]);pts;
*/ //ln&exp 1e5 uoj 766ms
   //ln 222ms
   //exp 568ms
    return 0;
}
```