神秘模板库

Toy ASM Truck

Huazhong University of Science and Technology

March 30, 2021

Contents

一切的开始

宏定义

数据结构

ST 表

- 二维

```
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。

```
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 $h(a,b,c,n)=nm(m+1)-2g(c,c-b-1,a,m-1)-2f(c,c-b-1,a,m-1)-f(a,b,c,n)$。

```cpp
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

```cpp
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;
}
```

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

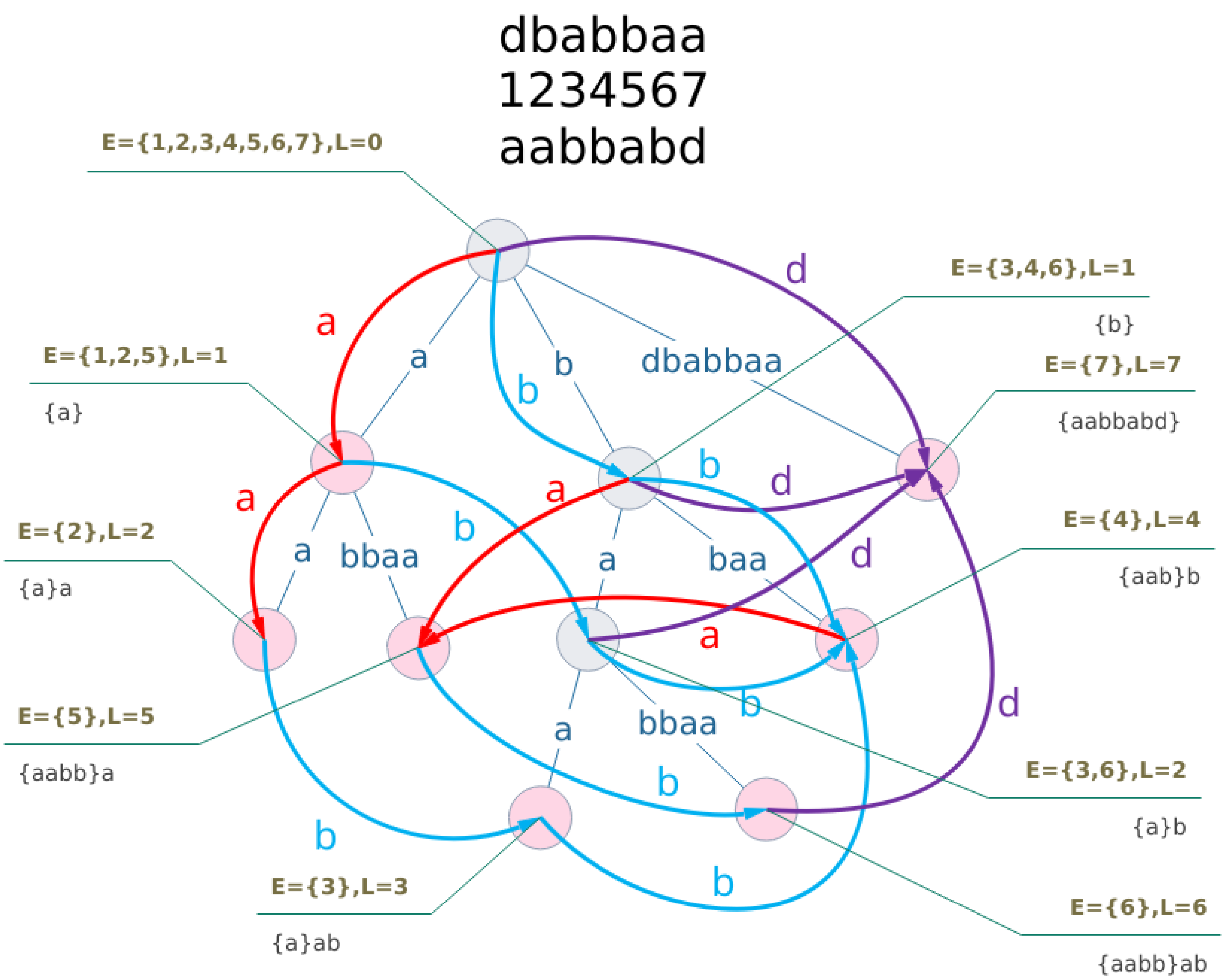
二维几何：点与向量

```
#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); }
// --------------------------------------------
```

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;
}
```