

多项式全家桶

```
1  #include <bits/stdc++.h>
2
3  #define I inline
4  #define fi first
5  #define se second
6  #define LL long long
7  #define mp make_pair
8  #define reg register int
9  #define pii pair<int,int>
10 #define fo(i, a, b) for(reg i = a; i <= b; i++)
11 #define fd(i, a, b) for(reg i = a; i >= b; i--)
12 #define ULL unsigned long long
13 #define cr const reg&
14 using namespace std;
15 const int inf = 2147483647;
16 const int mod = 998244353;
17 const int N = 1 << 17;
18
19 I int _max(cr x, cr y) {return x > y ? x : y;}
20 I int _min(cr x, cr y) {return x < y ? x : y;}
21 I LL read() {
22     LL x = 0, f = 1; char ch = getchar();
23     while(ch < '0' || ch > '9') {if(ch == '-') f = -1; ch = getchar();}
24     while(ch >= '0' && ch <= '9') x = (x << 3) + (x << 1) + (ch ^ 48), ch =
getchar();
25     return x * f;
26 }
27 I void ptt(LL x) {if(x >= 10) ptt(x / 10); putchar(x % 10 + '0');}
28 I void put(LL x) {x < 0 ? putchar('-'), ptt(-x) : ptt(x);}
29 I void pr1(LL x) {put(x), putchar(' ');}
30 I void pr2(LL x) {put(x), puts("");}
31
32 I int pow_mod(reg a, reg k) {reg ans = 1; for(; k; k >>= 1, a = (LL)a * a % mod)
if(k & 1) ans = (LL)ans * a % mod; return ans;}
33
34 namespace Poly {
35     int jc[N], inv[N], iv[N];
36     int w[N << 1], R[N << 1], h[N << 1]; ULL p[N << 1];
37     int cnt; vector<int> q[N << 2], o[N << 2];
38
39     I int Pre(cr n) {
40         reg len = 1; for(; len <= n; len <= 1);
41         iv[0] = iv[1] = 1; fo(i, 2, n) iv[i] = (LL)iv[mod % i] * (mod - mod / i) %
mod;
42         jc[0] = 1; fo(i, 1, n) jc[i] = (LL)jc[i - 1] * i % mod;
43         inv[n] = pow_mod(jc[n], mod - 2); fd(i, n, 1) inv[i - 1] = (LL)inv[i] * i %
mod;
44         for(reg i = 1; i < (len << 1); i <= 1) {
```

```

45         reg s = 1, wn = pow_mod(3, (mod - 1) / (i << 1));
46         fo(j, 0, i - 1) w[i + j] = s, s = (LL)s * wn % mod;
47     } return len;
48 }
49 I int pre(cr n) {
50     reg len = 1; for(; len <= n; len <<= 1);
51     fo(i, 0, len - 1) R[i] = (R[i >> 1] >> 1) | ((i & 1) ? (len >> 1) : 0);
52     return len;
53 }
54 I int gao(int x) {return x < 0 ? x + mod : x;}
55 I void DFT(int y[], cr len) {
56     fo(i, 0, len - 1) p[R[i]] = gao(y[i]);
57     int b;
58     for(reg i = 1; i < len; i <<= 1) for(reg j = 0; j < len; j += i << 1)
59         fo(k, 0, i - 1) b = p[j + k + i] * w[i + k] % mod, p[j + k + i] = p[j +
60 k] + mod - b, p[j + k] = p[j + k] + b;
61     fo(i, 0, len - 1) y[i] = p[i] % mod;
62 }
63 I void IDFT(int y[], cr len) {
64     reverse(y + 1, y + len); DFT(y, len); reg hh = pow_mod(len, mod - 2);
65     fo(i, 0, len - 1) y[i] = (LL)y[i] * hh % mod;
66 }
67 I void clear(int y[], cr len) {fo(i, len, (len << 1) - 1) y[i] = 0;}
68 I void clear(int a[], cr s, cr t) {if(s >= t) return; memset(a + s, 0,
69 sizeof(a[0]) * (t - s));}
70 I void cpy(int a[], int b[], cr len) {memcpy(a, b, sizeof(a[0]) * len), memset(a
71 + len, 0, sizeof(a[0]) * len);}
72 I void cpy(int a[], vector<int> b) {fo(i, 0, (int)b.size() - 1) a[i] = b[i];}
73 I void cpy(vector<int> &a, int b[], cr len) {a.resize(len); fo(i, 0, len - 1)
74 a[i] = b[i];}
75
76 I void getinv(int a[], int b[], cr len) {
77     if(len == 1) {b[0] = pow_mod(a[0], mod - 2), b[1] = 0; return;}
78     getinv(a, b, len >> 1); cpy(h, a, len), clear(b, len);
79     fo(i, 0, (len << 1) - 1) R[i] = (R[i >> 1] >> 1) | ((i & 1) ? len : 0);
80     DFT(h, len << 1), DFT(b, len << 1);
81     fo(i, 0, (len << 1) - 1) b[i] = (2 - (LL)b[i] * h[i]) % mod * b[i] % mod;
82     IDFT(b, len << 1); clear(b, len);
83 }
84 I void getln(int a[], int b[], cr len) {
85     getinv(a, b, len);
86     fo(i, 1, len - 1) h[i - 1] = (LL)a[i] * i % mod; h[len - 1] = 0; clear(h,
87 len);
88     DFT(h, len << 1), DFT(b, len << 1); fo(i, 0, (len << 1) - 1) b[i] = (LL)h[i]
89 * b[i] % mod;
90     IDFT(b, len << 1); fd(i, len - 1, 1) b[i] = (LL)b[i - 1] * iv[i] % mod;
91     clear(b, len), b[0] = 0;
92 }
93
94 const int Z = 16;
95 int MEM1[N * 10], z[N << 1]; ULL MEM2[N * 10], g[N << 1];
96 I void solve(cr l, cr r, int *MEM1, ULL *MEM2) {
97     if(r - l + 1 <= 64) {
98         fo(i, l, r) g[i] %= mod;
99     }
100 }

```

```

91     fo(i, 1, r) {
92         if(i == 0) g[i] = 1;
93         else g[i] = g[i] % mod * iv[i] % mod;
94         fo(j, i + 1, r) g[j] += g[i] * z[j - i];
95         if(i & 15) fo(j, i + 1, r) g[j] %= mod;
96     } return ;
97 } reg len = (r - 1 + 1) / Z, ll = len << 1;
98 fo(i, 0, ll - 1) R[i] = (R[i >> 1] >> 1) | ((i & 1) ? (ll >> 1) : 0);
99 int *h1[Z]; ULL *h2[Z];
100 fo(i, 0, Z - 1) {
101     h1[i] = MEMP1; MEMP1 += ll;
102     h2[i] = MEMP2; MEMP2 += ll;
103     fo(j, 0, ll - 1) h2[i][j] = 0;
104 } if(ll == 0) {
105     fo(i, 0, Z - 2) {
106         fo(j, 0, ll - 1) h1[i][j] = z[j + i * len];
107         DFT(h1[i], ll);
108     }
109 } fo(i, 0, Z - 1) {
110     fo(j, 0, ll - 1) h1[Z - 1][j] = h2[i][j] % mod;
111     IDFT(h1[Z - 1], ll);
112     fo(j, 0, len - 1) g[l + i * len + j] += h1[Z - 1][j + len];
113     solve(l + i * len, l + (i + 1) * len - 1, MEMP1, MEMP2);
114     if(i == Z - 1) return ;
115     fo(j, 0, ll - 1) R[j] = (R[j >> 1] >> 1) | ((j & 1) ? (ll >> 1) : 0),
h1[Z - 1][j] = 0;
116     fo(j, 0, len - 1) h1[Z - 1][j] = g[l + i * len + j];
117     DFT(h1[Z - 1], ll);
118     fo(j, i + 1, Z - 1) fo(k, 0, ll - 1) h2[j][k] += (LL)h1[Z - 1][k] * h1[j
- i - 1][k];
119 }
120 }
121 I void getexp(int a[], int b[], cr len) {
122     fo(i, 0, len - 1) z[i] = (LL)i * a[i] % mod, g[i] = 0;
123     solve(0, len - 1, MEM1, MEM2);
124     fo(i, 0, len - 1) b[i] = g[i];
125 }
126 int C[N << 1];
127 I void getpw(int a[], int b[], cr k, cr len) {
128     getln(a, C, len);
129     fo(i, 0, len - 1) C[i] = (LL)C[i] * k % mod;
130     getexp(C, b, len);
131 }
132 int V[N << 1];
133 I void mul(int a[], int b[], cr len1, cr len2) {
134     reg len = pre(len1 + len2 - 1);
135     if(len1 + len2 - 1 <= 64) {
136         clear(V, 0, len1 + len2 - 1);
137         fo(i, 0, len1 - 1) fo(j, 0, len2 - 1) v[i + j] = (V[i + j] + (LL)a[i] *
b[j]) % mod;
138         cpy(a, V, len1 + len2 - 1), clear(a + len1 + len2 - 1, len);
139         return ;
140     }

```

```

141     cpy(V, b, len2), clear(a, len1, len), clear(V, len2, len);
142     DFT(a, len), DFT(V, len);
143     fo(i, 0, len - 1) a[i] = (LL)a[i] * V[i] % mod;
144     IDFT(a, len); clear(a, len1 + len2 - 1, len);
145 }
146 I void MULT(int a[], int b[], cr len1, cr len2) {
147     reg len = pre(len1);
148     if(len1 <= 64) {
149         clear(V, 0, len1 - len2 + 1);
150         fo(i, 0, len1 - len2) {
151             fo(j, 0, min(len2 - 1, len1 - i - 1)) V[i] = (V[i] + (LL)a[i + j] *
b[j]) % mod;
152         } cpy(a, V, len1 - len2 + 1), clear(a, len1 - len2 + 1, len);
153         return ;
154     }
155     cpy(V, b, len2), reverse(V, V + len2), clear(V, len2, len), clear(a, len1,
len);
156     DFT(a, len), DFT(V, len);
157     fo(i, 0, len - 1) a[i] = (LL)a[i] * V[i] % mod;
158     IDFT(a, len); fo(i, 0, len1 - len2) a[i] = a[i + len2 - 1]; clear(a, len1 -
len2 + 1, len);
159 }
160 int h1[N << 1], h2[N << 1];
161 I void getdiv(int a[], int b[], int c[], cr n, cr m) {
162     reg ln = n - m + 1, len = pre(ln); cpy(h1, a, n), cpy(h2, b, m);
163     reverse(h1, h1 + n), reverse(h2, h2 + m), clear(h2, m, len);
164     getinv(h2, c, len), mul(c, h1, ln, ln), reverse(c, c + ln);
165 }
166 I void getmod(int a[], int b[], int c[], cr n, cr m) {
167     if(n < m) return ;
168     getdiv(a, b, c, n, m);
169     mul(c, b, n - m + 1, m);
170     fo(i, 0, n - 1) c[i] = (a[i] - c[i]) % mod;
171 }
172 vector<int> G[N << 1], F[N << 1]; int d[N], A[N << 1], B[N << 1];
173 I void gao1(cr u, cr l, cr r) {
174     if(l == r) {
175         G[u].push_back(1), G[u].push_back(gao(-d[l]));
176         G[u].resize(2);
177         return ;
178     } reg mid = l + r >> 1;
179     gao1(u << 1, l, mid), gao1(u << 1 | 1, mid + 1, r);
180     cpy(A, G[u << 1]), cpy(B, G[u << 1 | 1]);
181     mul(A, B, mid - l + 2, r - mid + 1);
182     cpy(G[u], A, r - l + 2);
183 }
184 I void gao2(cr u, cr l, cr r) {
185     if(l == r) {d[l] = F[u][0]; return ;}
186     reg mid = l + r >> 1;
187     cpy(A, F[u]), cpy(B, G[u << 1 | 1]);
188     MULT(A, B, r - l + 2, r - mid + 1);
189     cpy(F[u << 1], A, mid - l + 2);
190     cpy(A, F[u]), cpy(B, G[u << 1]);

```

```

191     MULT(A, B, r - 1 + 2, mid - 1 + 2);
192     cpy(F[u << 1 | 1], A, r - mid + 1);
193     gao2(u << 1, 1, mid), gao2(u << 1 | 1, mid + 1, r);
194 }
195 I void getval(int a[], int b[], cr n, cr m) {
196     fo(i, 1, m) d[i] = b[i];
197     gao1(1, 1, m);
198     cpy(A, G[1]);
199     reg len = pre(n + 1);
200     getinv(A, B, len); reverse(B, B + n + 1);
201     fo(i, 0, n) A[i] = a[i];
202     mul(A, B, n + 1, n + 1);
203     F[1].resize(m + 1);
204     fo(i, 0, m) F[1][i] = A[i + n];
205     gao2(1, 1, m);
206     fo(i, 1, m) b[i] = d[i];
207 }
208 }
209
210 int main() {
211     return 0;
212 }

```

分治FFT

```

1  #include <bits/stdc++.h>
2
3  #define I inline
4  #define fi first
5  #define se second
6  #define LL long long
7  #define mp make_pair
8  #define reg register int
9  #define pii pair<int,int>
10 #define fo(i, a, b) for(int i = a; i <= b; i++)
11 #define fd(i, a, b) for(reg i = a; i >= b; i--)
12 #define ULL unsigned long long
13 #define cr const reg&
14 using namespace std;
15 const int inf = 2147483647;
16 const int mod = 998244353;
17 const int N = 1 << 18;
18 const int B = 16;
19
20 I int _max(cr x, cr y) {return x > y ? x : y;}
21 I int _min(cr x, cr y) {return x < y ? x : y;}
22 I LL read() {
23     LL x = 0, f = 1; char ch = getchar();
24     while(ch < '0' || ch > '9') {if(ch == '-') f = -1; ch = getchar();}
25     while(ch >= '0' && ch <= '9') x = (x << 3) + (x << 1) + (ch ^ 48), ch =
getchar();
26     return x * f;

```

```

27 }
28 I void ptt(LL x) {if(x >= 10) ptt(x / 10); putchar(x % 10 + '0');}
29 I void put(LL x) {x < 0 ? putchar('-'), ptt(-x) : ptt(x);}
30 I void pr1(LL x) {put(x), putchar(' ');}
31 I void pr2(LL x) {put(x), puts("");}
32
33 I int pow_mod(reg a, reg k) {reg ans = 1; for(; k >= 1, a = (LL)a * a % mod)
    if(k & 1) ans = (LL)ans * a % mod; return ans;}
34
35 ULL p[N << 1], g[N], f[N];
36 int A[N << 1], R[N << 1], w[N << 1], jc[N], inv[N], iv[N];
37 int MEM1[N * 10], MEM2[N * 10], MEM3[N * 10], MEM4[N * 10];
38
39 I int Pre(cr n) {
40     reg len = 1; for(; len <= n; len <<= 1);
41     jc[0] = 1; fo(i, 1, n) jc[i] = (LL)jc[i - 1] * i % mod;
42     inv[n] = pow_mod(jc[n], mod - 2); fd(i, n, 1) inv[i - 1] = (LL)inv[i] * i % mod;
43     iv[0] = iv[1] = 1; fo(i, 2, n) iv[i] = (LL)iv[mod % i] * (mod - mod / i) % mod;
44     for(reg i = 1; i < len; i <<= 1) {
45         reg wn = pow_mod(3, (mod - 1) / (i << 1)), s = 1;
46         fo(k, 0, i - 1) w[i + k] = s, s = (LL)s * wn % mod;
47     } return len;
48 }
49 I int pre(cr n) {
50     reg len = 1; for(; len <= n; len <<= 1);
51     fo(i, 0, len - 1) R[i] = (R[i >> 1] >> 1) | ((i & 1) ? (len >> 1) : 0);
52     return len;
53 }
54
55 I void DFT(int y[], cr len) {
56     fo(i, 0, len - 1) p[R[i]] = y[i]; reg b;
57     for(reg i = 1; i < len; i <<= 1) for(reg j = 0; j < len; j += i << 1)
58         fo(k, 0, i - 1) b = p[i + j + k] * w[i + k] % mod, p[i + j + k] = p[j + k] -
b + mod, p[j + k] = p[j + k] + b;
59     fo(i, 0, len - 1) y[i] = p[i] % mod;
60 }
61 I void IDFT(int y[], cr len) {
62     reverse(y + 1, y + len); DFT(y, len);
63     reg hh = pow_mod(len, mod - 2); fo(i, 0, len - 1) y[i] = (LL)y[i] * hh % mod;
64 }
65
66 I void solve(cr l, cr r, int *MEM1, int *MEM2, int *MEM3, int *MEM4) {
67     if(r - l + 1 <= 64) {
68         fo(i, l, r) {
69             if(i == 0) f[0] = g[0] = 0;
70             else if(i == 1) f[i] = 1;
71             else f[i] = (LL)f[i] * inv[i] % mod * iv[2] % mod;
72             if(l == 0) fo(j, l, i - 1) g[i] = (g[i] + (LL)f[j] * f[i - j]) % mod;
73             else fo(j, l, i - 1) g[i] = (g[i] + 2LL * f[j] * f[i - j]) % mod;
74             fo(j, i + 1, r) f[j] = (f[j] + g[i] * A[j - i]) % mod;
75         } return ;
76     } int len = (r - l + 1) / B, ll = len << 1; pre(ll - 1);
77     int *h1[B], *h2[B], *h3[B], *h4[B];

```

```

78     fo(i, 0, B - 1) {
79         h1[i] = MEMP1, MEMP1 += 11;
80         h2[i] = MEMP2, MEMP2 += 11;
81         h3[i] = MEMP3, MEMP3 += 11;
82         h4[i] = MEMP4, MEMP4 += 11;
83         fo(j, 0, 11 - 1) h2[i][j] = h4[i][j] = 0;
84     } if(l == 0) {
85         fo(i, 0, B - 2) {
86             fo(j, 0, 11 - 1) h1[i][j] = A[j + i * len];
87             DFT(h1[i], 11);
88         }
89     } fo(i, 0, B - 1) {
90         IDFT(h2[i], 11);
91         fo(j, 0, len - 1) f[l + i * len + j] = (f[l + i * len + j] + h2[i][j + len])
% mod;
92         fo(j, 0, 11 - 1) h1[B - 1][j] = 0;
93         if(l == 0) fo(j, 0, i - 1) fo(k, 0, 11 - 1) h1[B - 1][k] = (h1[B - 1][k] +
(LL)h3[i - j - 1][k] * h4[j][k]) % mod;
94         else fo(j, 0, i - 1) fo(k, 0, 11 - 1) h1[B - 1][k] = (h1[B - 1][k] + 2LL *
h3[i - j - 1][k] * h4[j][k]) % mod;
95         IDFT(h1[B - 1], 11);
96         fo(j, 0, len - 1) g[l + i * len + j] = (g[l + i * len + j] + h1[B - 1][j +
len]) % mod;
97         solve(l + i * len, l + (i + 1) * len - 1, MEMP1, MEMP2, MEMP3, MEMP4);
98         if(i == B - 1) break;
99         pre(11 - 1);
100        if(l == 0) {
101            fo(j, 0, 11 - 1) h3[i][j] = h4[i][j] = 0;
102            fo(j, 0, len - 1) h3[i][j] = h4[i][j] = f[l + i * len + j];
103            if(i) {
104                IDFT(h3[i - 1], 11);
105                fo(j, 0, len - 1) h3[i - 1][j + len] = f[l + i * len + j];
106                DFT(h3[i - 1], 11);
107            } DFT(h3[i], 11), DFT(h4[i], 11);
108        } else {
109            fo(j, 0, 11 - 1) h4[i][j] = 0;
110            fo(j, 0, len - 1) h4[i][j] = f[l + i * len + j];
111            DFT(h4[i], 11);
112        } fo(j, 0, 11 - 1) h1[B - 1][j] = 0;
113        fo(j, 0, len - 1) h1[B - 1][j] = g[l + i * len + j];
114        DFT(h1[B - 1], 11);
115        fo(j, i + 1, B - 1) fo(k, 0, 11 - 1) h2[j][k] = (h2[j][k] + (LL)h1[B - 1][k]
* h1[j - i - 1][k]) % mod;
116    } pre(11 - 1); if(l == 0) {
117        IDFT(h3[B - 2], 11);
118        fo(j, 0, len - 1) h3[B - 2][j + len] = f[l + (B - 1) * len + j];
119        DFT(h3[B - 2], 11);
120    }
121 }
122
123 char ss[N];
124
125 int main() {

```

```

126     reg n = read(), len = Pre(n);
127     scanf("%s", ss);
128     fo(i, 0, n - 1) A[i] = ss[i] - '0';
129     fd(i, n - 1, 0) A[i + 1] = A[i] * inv[i]; A[0] = 0;
130     solve(0, len - 1, MEM1, MEM2, MEM3, MEM4);
131     fo(i, 1, n) pr2((LL)f[i] * jc[i] % mod);
132     return 0;
133 }

```

BM算法

```

1  #include <bits/stdc++.h>
2
3  #define I inline
4  #define fi first
5  #define se second
6  #define LL long long
7  #define mp make_pair
8  #define reg register int
9  #define pii pair<int,int>
10 #define fo(i, a, b) for(reg i = a; i <= b; i++)
11 #define fd(i, a, b) for(reg i = a; i >= b; i--)
12 #define ULL unsigned long long
13 #define cr const reg&
14 using namespace std;
15 const int inf = 2147483647;
16 const int mod = 998244353;
17 const int N = 2e4 + 1;
18
19 I int _max(cr x, cr y) {return x > y ? x : y;}
20 I int _min(cr x, cr y) {return x < y ? x : y;}
21 I LL read() {
22     LL x = 0, f = 1; char ch = getchar();
23     while(ch < '0' || ch > '9') {if(ch == '-') f = -1; ch = getchar();}
24     while(ch >= '0' && ch <= '9') x = (x << 3) + (x << 1) + (ch ^ 48), ch =
getchar();
25     return x * f;
26 }
27 I void ptt(LL x) {if(x >= 10) ptt(x / 10); putchar(x % 10 + '0');}
28 I void put(LL x) {x < 0 ? putchar('-') : ptt(x);}
29 I void pr1(LL x) {put(x), putchar(' ');}
30 I void pr2(LL x) {put(x), puts("");}
31
32 I int pow_mod(reg a, reg k) {reg ans = 1; for(; k; k >>= 1, a = (LL)a * a % mod)
if(k & 1) ans = (LL)ans * a % mod; return ans;}
33
34 namespace Poly {
35     int jc[N], inv[N], iv[N];
36     int w[N << 1], R[N << 1], h[N << 1]; ULL p[N << 1];
37     int cnt; vector<int> q[N << 2], o[N << 2];

```



```

38
39 I int Pre(cr n) {
40     reg len = 1; for(; len <= n; len <= 1);
41     jc[0] = 1; fo(i, 1, n) jc[i] = (LL)jc[i - 1] * i % mod;
42     inv[n] = pow_mod(jc[n], mod - 2); fd(i, n, 1) inv[i - 1] = (LL)inv[i] * i %
mod;
43     for(reg i = 1; i < (len <= 1); i <= 1) {
44         reg s = 1, wn = pow_mod(3, (mod - 1) / (i <= 1));
45         fo(j, 0, i - 1) w[i + j] = s, s = (LL)s * wn % mod;
46     } return len;
47 }
48 I int pre(cr n) {
49     reg len = 1; for(; len <= n; len <= 1);
50     fo(i, 0, len - 1) R[i] = (R[i >> 1] >> 1) | ((i & 1) ? (len >> 1) : 0);
51     return len;
52 }
53 I int gao(int x) {return x < 0 ? x + mod : x;}
54 I void DFT(int y[], cr len) {
55     fo(i, 0, len - 1) p[R[i]] = gao(y[i]);
56     int b;
57     for(reg i = 1; i < len; i <= 1) for(reg j = 0; j < len; j += i <= 1)
58         fo(k, 0, i - 1) b = p[j + k + i] * w[i + k] % mod, p[j + k + i] = p[j +
k] + mod - b, p[j + k] = p[j + k] + b;
59     fo(i, 0, len - 1) y[i] = p[i] % mod;
60 }
61 I void IDFT(int y[], cr len) {
62     reverse(y + 1, y + len); DFT(y, len); reg hh = pow_mod(len, mod - 2);
63     fo(i, 0, len - 1) y[i] = (LL)y[i] * hh % mod;
64 }
65 I void clear(int y[], cr len) {fo(i, len, (len <= 1) - 1) y[i] = 0;}
66 I void clear(int a[], cr s, cr t) {if(s >= t) return ; memset(a + s, 0,
sizeof(a[0]) * (t - s));}
67 I void cpy(int a[], int b[], cr len) {memcpy(a, b, sizeof(a[0]) * len), memset(a
+ len, 0, sizeof(a[0]) * len);}
68 I void cpy(int a[], vector<int> b) {fo(i, 0, (int)b.size() - 1) a[i] = b[i];}
69 I void cpy(vector<int> &a, int b[], cr len) {a.resize(len); fo(i, 0, len - 1)
a[i] = b[i];}
70
71 I void getinv(int a[], int b[], cr len) {
72     if(len == 1) {b[0] = pow_mod(a[0], mod - 2), b[1] = 0; return ;}
73     getinv(a, b, len >> 1); cpy(h, a, len), clear(b, len);
74     fo(i, 0, (len <= 1) - 1) R[i] = (R[i >> 1] >> 1) | ((i & 1) ? len : 0);
75     DFT(h, len <= 1), DFT(b, len <= 1);
76     fo(i, 0, (len <= 1) - 1) b[i] = (2 - (LL)b[i] * h[i]) % mod * b[i] % mod;
77     IDFT(b, len <= 1); clear(b, len);
78 }
79 I void getln(int a[], int b[], cr len) {
80     getinv(a, b, len);
81     fo(i, 1, len - 1) h[i - 1] = (LL)a[i] * i % mod; h[len - 1] = 0; clear(h,
len);
82     DFT(h, len <= 1), DFT(b, len <= 1); fo(i, 0, (len <= 1) - 1) b[i] = (LL)h[i]
* b[i] % mod;

```

```

83     IDFT(b, len << 1); fd(i, len - 1, 1) b[i] = (LL)b[i - 1] * iv[i] % mod;
clear(b, len), b[0] = 0;
84 }
85 const int B = 16;
86 int MEM1[N * 10], z[N << 1]; ULL MEM2[N * 10], g[N << 1];
87 I void solve(cr l, cr r, int *MEMP1, ULL *MEMP2) {
88     if(r - l + 1 <= 64) {
89         fo(i, l, r) g[i] %= mod;
90         fo(i, l, r) {
91             if(i == 0) g[i] = 1;
92             else g[i] = g[i] % mod * iv[i] % mod;
93             fo(j, i + 1, r) g[j] += g[i] * z[j - i];
94             if(i & 15) fo(j, i + 1, r) g[j] %= mod;
95         } return ;
96     } reg len = (r - l + 1) / B, ll = len << 1;
97     fo(i, 0, ll - 1) R[i] = (R[i >> 1] >> 1) | ((i & 1) ? (ll >> 1) : 0);
98     int *h1[B]; ULL *h2[B];
99     fo(i, 0, B - 1) {
100         h1[i] = MEMP1; MEMP1 += ll;
101         h2[i] = MEMP2; MEMP2 += ll;
102         fo(j, 0, ll - 1) h2[i][j] = 0;
103     } if(l == 0) {
104         fo(i, 0, B - 2) {
105             fo(j, 0, ll - 1) h1[i][j] = z[j + i * len];
106             DFT(h1[i], ll);
107         }
108     } fo(i, 0, B - 1) {
109         fo(j, 0, ll - 1) h1[B - 1][j] = h2[i][j] % mod;
110         IDFT(h1[B - 1], ll);
111         fo(j, 0, len - 1) g[l + i * len + j] += h1[B - 1][j + len];
112         solve(l + i * len, l + (i + 1) * len - 1, MEMP1, MEMP2);
113         if(i == B - 1) return ;
114         fo(j, 0, ll - 1) R[j] = (R[j >> 1] >> 1) | ((j & 1) ? (ll >> 1) : 0),
h1[B - 1][j] = 0;
115         fo(j, 0, len - 1) h1[B - 1][j] = g[l + i * len + j];
116         DFT(h1[B - 1], ll);
117         fo(j, i + 1, B - 1) fo(k, 0, ll - 1) h2[j][k] += (LL)h1[B - 1][k] * h1[j
- i - 1][k];
118     }
119 }
120 I void getexp(int a[], int b[], cr len) {
121     fo(i, 0, len - 1) z[i] = (LL)i * a[i] % mod;
122     solve(0, len - 1, MEM1, MEM2);
123     fo(i, 0, len - 1) b[i] = g[i];
124 }
125 int V[N << 1];
126 I void mul(int a[], int b[], cr len1, cr len2) {
127     reg len = pre(len1 + len2 - 1);
128     cpy(V, b, len2), clear(a, len1, len), clear(V, len2, len);
129     DFT(a, len), DFT(V, len);
130     fo(i, 0, len - 1) a[i] = (LL)a[i] * V[i] % mod;
131     IDFT(a, len);
132 }

```

```

133     int h1[N << 1], h2[N << 1];
134     I void getdiv(int a[], int b[], int c[], cr n, cr m) {
135         reg ln = n - m + 1, len = pre(ln); cpy(h1, a, n), cpy(h2, b, m);
136         reverse(h1, h1 + n), reverse(h2, h2 + m), clear(h2, m, len);
137         getinv(h2, c, len), mul(c, h1, ln, ln), reverse(c, c + ln);
138     }
139     I void getmod(int a[], int b[], int c[], cr n, cr m) {
140         if(n < m) return ;
141         getdiv(a, b, c, n, m);
142         mul(c, b, n - m + 1, m);
143         fo(i, 0, n - 1) c[i] = (a[i] - c[i]) % mod;
144     }
145     int t[N], ans[N];
146     I void pre(cr u, cr l, cr r) {
147         q[u].clear();
148         if(r - l + 1 <= 64) {
149             q[u].resize(r - l + 2); q[u][0] = 1;
150             fo(i, l, r) fd(j, i - 1, 0) q[u][j + 1] = (q[u][j + 1] + q[u][j]) % mod,
q[u][j] = (LL)q[u][j] * (mod - t[i]) % mod;
151             return ;
152         } reg mid = l + r >> 1, lc = u << 1, rc = u << 1 | 1;
153         pre(lc, l, mid), pre(rc, mid + 1, r);
154         reg l1 = q[lc].size(), l2 = q[rc].size(); cpy(h1, q[lc]), cpy(h2, q[rc]);
155         mul(h1, h2, l1, l2); cpy(q[u], h1, l1 + l2 - 1);
156     }
157     int x[N << 1], y[N << 1], z[N << 1];
158     I void sol(cr u, cr l, cr r) {
159         if(r - l + 1 <= 64) {
160             fo(i, l, r) {
161                 reg z = 1;
162                 fo(j, 0, (int)o[u].size() - 1) ans[i] = (ans[i] + (LL)o[u][j] * z) %
mod, z = (LL)z * t[i] % mod;
163             } return ;
164         } reg mid = l + r >> 1, lc = u << 1, rc = u << 1 | 1;
165         reg l1 = (int)o[u].size(), l1 = (int)q[lc].size(), l2 = (int)q[rc].size(),
ln;
166         cpy(x, o[u]), cpy(y, q[lc]); getmod(x, y, z, l1, l1);
167         ln = l1; while(ln > 1 && !Z[ln - 1]) ln--; cpy(o[lc], z, ln);
168         cpy(x, o[u]), cpy(y, q[rc]); getmod(x, y, z, l1, l2);
169         ln = l2; while(ln > 1 && !Z[ln - 1]) ln--; cpy(o[rc], z, ln);
170         sol(lc, l, mid), sol(rc, mid + 1, r);
171     }
172     I void getval(int a[], int b[], cr n, cr m) {
173         fo(i, 1, m) t[i] = b[i];
174         pre(1, 1, m); cpy(o[1], a, n);
175         sol(1, 1, m);
176         fo(i, 1, m) b[i] = ans[i] < 0 ? ans[i] + mod : ans[i];
177     }
178 }
179 int l1, mo[N];
180 namespace BM {
181     int f[3][N], len[3], fail[3], ad[3];
182     I void cpy(cr u, cr v) {

```

```

183     fo(i, 1, len[v]) f[v][i] = 0;
184     fo(i, 1, len[u]) f[v][i] = f[u][i]; len[v] = len[u], fail[v] = fail[u],
ad[v] = ad[u];
185 }
186 I void main(cr n, int a[]) {
187     fo(i, 0, n - 1) {
188         reg v = a[i];
189         fo(j, 1, len[0]) v = (v - (LL)f[0][j] * a[i - j]) % mod;
190         if(!v) continue;
191         ad[0] = v, fail[0] = i;
192         if(!len[0]) {cpy(0, 1), len[0] = 1; continue;}
193         cpy(0, 2); reg hh = (LL)v * pow_mod(ad[1], mod - 2) % mod;
194         reg st = i - fail[1]; len[0] = _max(len[0], len[1] + st);
195         f[0][st] = (f[0][st] + hh) % mod;
196         fo(j, 1, fail[1]) f[0][j + st] = (f[0][j + st] - (LL)f[1][j] * hh) %
mod;
197         if(len[2] - fail[2] < len[1] - fail[1]) cpy(2, 1);
198     } fo(i, 1, len[0]) mo[i] = f[0][i]; ll = len[0];
199 }
200 }
201 int p[N], ans[N], nw[N], c[N];
202
203 I void mul(int a[], int b[]) {
204     Poly::mul(a, b, ll, ll);
205     Poly::getmod(a, mo, c, ll * 2 - 1, ll + 1);
206     fo(i, 0, ll - 1) a[i] = c[i];
207 }
208
209 int main() {
210     reg n = read(), m = read();
211     fo(i, 0, n - 1) p[i] = read();
212     BM::main(n, p);
213     fo(i, 1, ll) pr1(mo[i] < 0 ? mo[i] + mod : mo[i]); puts("");
214     reverse(mo + 1, mo + ll + 1);
215     fo(i, 1, ll) mo[i - 1] = (mod - mo[i]) % mod; mo[ll] = 1;
216     Poly::Pre(ll * 2);
217     ans[0] = 1, nw[1] = 1;
218     for(; m; m >>= 1, mul(nw, nw)) if(m & 1) mul(ans, nw);
219     reg sum = 0;
220     fo(i, 0, ll - 1) sum = (sum + (LL)ans[i] * p[i]) % mod;
221     pr2(sum < 0 ? sum + mod : sum);
222     return 0;
223 }
224

```

可持久化平衡树 (FHQ)

```

1
2 #include <bits/stdc++.h>
3
4 #define I inline
5 #define fi first

```

```

6  #define se second
7  #define R register
8  #define LL long long
9  #define mp make_pair
10 #define reg register int
11 #define pii pair<int,int>
12 #define fo(i, a, b) for(reg i = a; i <= b; i++)
13 #define fd(i, a, b) for(reg i = a; i >= b; i--)
14 #define cr const reg&
15 using namespace std;
16 const int inf = 2147483647;
17 const int N = 2e5 + 1;
18
19 I int _max(cr x, cr y) {return x > y ? x : y;}
20 I int _min(cr x, cr y) {return x < y ? x : y;}
21 I LL read() {
22     LL x = 0, f = 1; char ch = getchar();
23     while(ch < '0' || ch > '9') {if(ch == '-') f = -1; ch = getchar();}
24     while(ch >= '0' && ch <= '9') x = (x << 3) + (x << 1) + (ch ^ 48), ch =
getchar();
25     return x * f;
26 }
27 I void ptt(LL x) {if(x >= 10) ptt(x / 10); putchar(x % 10 + '0');}
28 I void put(LL x) {x < 0 ? putchar('-'), ptt(-x) : ptt(x);}
29 I void pr1(LL x) {put(x), putchar(' ');}
30 I void pr2(LL x) {put(x), puts("");}
31
32 struct tnode {
33     int cnt, lc[N * 70], rc[N * 70], tot[N * 70]; LL val[N * 70], sum[N * 70];
34     bool lzy[N * 70];
35
36     I void upd(cr u) {sum[u] = sum[lc[u]] + sum[rc[u]] + val[u], tot[u] = tot[lc[u]]
+ tot[rc[u]] + 1;}
37     I int copy(cr v) {reg u = ++cnt; lc[u] = lc[v], rc[u] = rc[v], val[u] = val[v],
tot[u] = tot[v], sum[u] = sum[v], lzy[u] = lzy[v]; return u;}
38     I void rev(cr u) {lzy[u] ^= 1, swap(lc[u], rc[u]);}
39     I void pushdown(cr u) {
40         if(!lzy[u]) return ;
41         if(lc[u]) lc[u] = copy(lc[u]), rev(lc[u]);
42         if(rc[u]) rc[u] = copy(rc[u]), rev(rc[u]);
43         lzy[u] = 0;
44     } I int newnode(LL v) {++cnt; val[cnt] = sum[cnt] = v, tot[cnt] = 1; return cnt;}
45     I void Split(int now, int k, int &u, int &v) {
46         if(!now) {u = v = 0; return ;}
47         pushdown(now);
48         if(tot[lc[now]] < k) u = copy(now), Split(rc[u], k - tot[lc[now]] - 1, rc[u],
v), upd(u);
49         else v = copy(now), Split(lc[v], k, u, lc[v]), upd(v);
50     }
51     I int Merge(cr u, cr v) {
52         if(!u || !v) return u | v;
53         pushdown(u), pushdown(v);

```

```

54         if(rand() % (tot[u] + tot[v]) < tot[u]) {rc[u] = Merge(rc[u], v), upd(u);
return u;}
55         else {lc[v] = Merge(u, lc[v]), upd(v); return v;}
56     }
57     I void ins(int &rt, LL k, LL v) {
58         int x, y; Split(rt, k, x, y);
59         rt = Merge(Merge(x, newnode(v)), y);
60     }
61     I void del(int &rt, LL k) {
62         int x, y, z; Split(rt, k, x, y), Split(x, k - 1, x, z);
63         rt = Merge(x, y);
64     }
65     I void rev(int &rt, LL l, LL r) {
66         int x, y, z; Split(rt, r, x, y), Split(x, l - 1, x, z);
67         rev(z); rt = Merge(Merge(x, z), y);
68     }
69     I LL getsum(int &rt, LL l, LL r) {
70         int x, y, z; Split(rt, r, x, y);
71         Split(x, l - 1, x, z);
72         LL s = sum[z]; rt = Merge(Merge(x, z), y);
73         return s;
74     }
75 } t; int rt[N];
76
77 int main() {
78     reg n = read();
79     LL lst = 0;
80     fo(i, 1, n) {
81         reg v = read(); rt[i] = rt[v];
82         reg o = read(); LL l = read() ^ lst, r;
83         if(o ^ 2) r = read() ^ lst;
84         if(o == 1) t.ins(rt[i], l, r);
85         else if(o == 2) t.del(rt[i], l);
86         else if(o == 3) t.rev(rt[i], l, r);
87         else pr2(lst = t.getsum(rt[i], l, r));
88     }
89     return 0;
90 }

```

莫队二次离线

```

1 // luogu-judger-enable-o2
2 #include <bits/stdc++.h>
3
4 #define LL long long
5 #define pii pair<int,int>
6 using namespace std;
7 const int inf = 2147483647;
8 const int N = 100001;
9
10 int _max(int x, int y) {return x > y ? x : y;}
11 int _min(int x, int y) {return x < y ? x : y;}

```

```

12 inline int read() {
13     int x = 0, f = 1; char ch = getchar();
14     while(ch < '0' || ch > '9') {if(ch == '-') f = -1; ch = getchar();}
15     while(ch >= '0' && ch <= '9') x = (x << 3) + (x << 1) + (ch ^ 48), ch =
getchar();
16     return x * f;
17 }
18 void put(LL x) {
19     if(x < 0) putchar('-'), x = -x;
20     if(x >= 10) put(x / 10);
21     putchar(x % 10 + '0');
22 }
23
24 struct query {
25     int ll, l, r, id;
26 } q[N]; vector<query> g[N];
27 int len, gg[N], a[N], sum[N];
28 LL lans[N], ans[N], zz[N];
29
30 bool cmp(query a, query b) {
31     if(a.ll == b.ll) {
32         if(a.r == b.r) return a.l < b.l;
33         else return a.r < b.r;
34     } return a.ll < b.ll;
35 }
36
37 int main() {
38     int n = read(), m = read(), k = read();
39     for(int i = 0; i < 16384; i++) {
40         int x = i, sum = 0;
41         for(; x; x -= (x & -x)) sum++;
42         if(sum == k) gg[++len] = i;
43     } for(int i = 1; i <= n; i++) a[i] = read();
44     int kk = sqrt(m);
45     for(int i = 1; i <= m; i++) q[i].l = read(), q[i].r = read(), q[i].id = i,
q[i].ll = (q[i].l - 1) / kk + 1, q[i].id = i;
46     sort(q + 1, q + m + 1, cmp);
47     int l = 1, r = 0;
48     for(int i = 1; i <= m; i++) {
49         if(r < q[i].r) g[l - 1].push_back(query{-1, r + 1, q[i].r, i});
50         else if(q[i].r < r) g[l - 1].push_back(query{1, q[i].r + 1, r, i});
51         r = q[i].r;
52         if(l < q[i].l) {
53             if(k == 0) ans[i] -= q[i].l - 1;
54             g[r].push_back(query{-1, l, q[i].l - 1, i});
55         } else if(q[i].l < l) {
56             if(k == 0) ans[i] += l - q[i].l;
57             g[r].push_back(query{1, q[i].l, l - 1, i});
58         } l = q[i].l;
59     } for(int i = 1; i <= n; i++) {
60         for(int j = 1; j <= len; j++) sum[a[i] ^ gg[j]]++;
61         zz[i] = zz[i - 1] + sum[a[i]];
62         for(int j = 0; j < g[i].size(); j++) {

```

```

63         int ll = g[i][j].l, rr = g[i][j].r, o = g[i][j].ll, id = g[i][j].id;
64         for(int j = ll; j <= rr; j++) ans[id] += o * sum[a[j]];
65     }
66     for(int i = 0; i <= n; i++) {
67         for(int j = 0; j < g[i].size(); j++) {
68             int ll = g[i][j].l, rr = g[i][j].r, o = -g[i][j].ll, id = g[i][j].id;
69             ans[id] += o * (zz[rr] - zz[ll - 1]);
70         }
71     } for(int i = 1; i <= m; i++) ans[i] += ans[i - 1], lans[q[i].id] = ans[i] - (k
== 0 ? q[i].r - q[i].l + 1 : 0);
72     for(int i = 1; i <= m; i++) put(lans[i]), puts("");
73     return 0;
74 }

```

BSGS

```

1  #include <cmath>
2  #include <cstdio>
3  #include <cstring>
4
5  using namespace std;
6  typedef long long LL;
7  const LL mod = 1000007;
8
9  int cnt;
10 LL head[1100000], next[110000], num[110000];
11
12 LL pow_mod(LL x, LL k, LL mod) {
13     x %= mod; LL ans = 1;
14     while(k) {
15         if(k % 2 == 1) (ans *= x) %= mod;
16         (x *= x) %= mod; k /= 2;
17     } return ans;
18 }
19
20 LL exgcd(LL a, LL b, LL &x, LL &y) {
21     if(b == 0) {
22         x = 1; y = 0;
23         return a;
24     } else {
25         LL tx, ty, d = exgcd(b, a % b, tx, ty);
26         x = ty; y = tx - ty * (a / b);
27         return d;
28     }
29 }
30
31 LL BSGS(LL a, LL b, LL p) {
32     LL t = sqrt(p) + 1, u = 1;
33     for(int i = 1; i <= t; i++) {
34         num[i] = (u * b) % p;
35         LL y = num[i] % mod;
36         if(!head[y]) head[y] = i;

```



```

37     else {
38         int o = head[y];
39         bool bk = 0;
40         while(next[o]) {
41             if(num[o] == num[i]) {bk = 1; break;}
42             o = next[o];
43         } if(!bk) next[o] = i;
44     }
45     (u *= a) %= p;
46 }
47 a = pow_mod(a, t, p); u = 1;
48 if(a == 0) {
49     if(b == 0) return 1;
50     return -1;
51 }
52 for(int i = 0; i < t; i++) {
53     LL y = u % mod;
54     int o = head[y];
55     while(num[o] != u && o) o = next[o];
56     if(num[o] == u) {
57         int j = o - 1;
58         if(i * t - j >= 0) return i * t - j;
59     } (u *= a) %= p;
60 } return -1;
61 }
62
63 int main() {
64     int tt, k; scanf("%d%d", &tt, &k);
65     while(tt--) {
66         LL y, z, p; scanf("%lld%lld%lld", &y, &z, &p);
67         memset(head, 0, sizeof(head));
68         memset(next, 0, sizeof(next));
69         memset(num, 0, sizeof(num));
70         if(k == 1) printf("%lld\n", pow_mod(y, z, p));
71         else if(k == 2) {
72             LL a = y, b = p, c = z;
73             LL x, y; LL d = exgcd(a, b, x, y);
74             if(c % d != 0) printf("Orz, I cannot find x!\n");
75             else {
76                 (x *= c / d) %= b;
77                 (x += b) %= b;
78                 printf("%lld\n", x);
79             }
80         } else {
81             LL ans = BSGS(y, z, p);
82             if(ans == -1) printf("Orz, I cannot find x!\n");
83             else printf("%lld\n", ans);
84         }
85     }
86     return 0;
87 }

```