```cpp
#include <bits/stdc++.h>

using namespace std;

template<typename T>
class AVL;

template<typename T>
void delete_avl(AVL<T> *p);

// f(l, r) -> l < r
template<typename T>
class AVL {
  T data;
  function<bool(T, T)> f;
  int size, height, balance;
  AVL<T> *left, *right;
public:
  AVL<T>(T data, function<bool(T, T)> f) : data(data), f(f) {
    size = 1;
    height = 1;
    balance = 0;
    left = right = nullptr;
  }

  void update() {
    int l_height = 0, r_height = 0, l_size = 0, r_size;
    if (left != nullptr) {
      l_height = left->height;
      l_size = left->size;
    }
    if (right != nullptr) {
      r_height = right->height;
      r_size = right->size;
    }
    height = max(l_height, r_height) + 1;
    size = 1 + l_size + r_size;
    balance = l_height - r_height;
  }

  void print() {
    cerr << "(";
    if (left != nullptr) left->print();
    cerr << data;
    if (right != nullptr) right->print();
    cerr << ")";
  }

  AVL<T> *rotate_r() {
    AVL<T> *l = left, *lr = l->right;
```

```
  51     l->right = this;
  52     left = lr;
  53     update();
  54     l->update();
  55     return l;
  56   }
  57
  58   AVL<T> *rotate_l() {
  59     AVL<T> *r = right, *rl = r->left;
  60     r->left = this;
  61     right = rl;
  62     update();
  63     r->update();
  64     return r;
  65   }
  66
  67   AVL<T> *insert(T val) {
  68     if (f(val, data)) {
  69       if (left == nullptr) {
  70         left = new AVL<T>(val, f);
  71       } else {
  72         left = left->insert(val);
  73       }
  74     } else {
  75       if (right == nullptr) {
  76         right = new AVL<T>(val, f);
  77       } else {
  78         right = right->insert(val);
  79       }
  80     }
  81     update();
  82     if (balance < -1) {
  83       if (right->balance == 1) right = right->rotate_r();
  84       return rotate_l();
  85     } else if (balance > 1) {
  86       if (left->balance == -1) left = left->rotate_l();
  87       return rotate_r();
  88     }
  89     return this;
  90   }
  91
  92   // バグってる
  93   AVL<T> *erase(T val) {
  94     if (f(val, data)) {
  95       if (left != nullptr) {
  96         left = left->erase(val);
  97       }
  98     } else if (f(data, val)) {
  99       if (right != nullptr) {
 100         right = right->erase(val);
```

```
101       }
102     } else {
103       // delete all
104       // delete_avl(this);
105
106       if (left != nullptr) {
107         left = left->erase(val);
108       } else if (right != nullptr) {
109         right = right->erase(val);
110       } else {
111         delete this;
112         return nullptr;
113       }
114     }
115     update();
116     if (balance < -1) {
117       if (right->balance == -1) right = right->rotate_l();
118       return rotate_l();
119     } else if (balance > 1) {
120       if (left->balance == 1) left = left->rotate_r();
121       return rotate_r();
122     }
123     return this;
124   }
125 };
126
127 template<typename T>
128 void delete_avl(AVL<T> *p) {
129   if (p == nullptr) return;
130   delete_avl<T>(p->left);
131   delete_avl<T>(p->right);
132   delete p;
133 }
134
135 using avl = AVL<int>;
136
137 int main() {
138   avl *root = new avl(5, [](int l, int r) {
139     return l < r;
140   });
141   for (int i = 0; i < 10; i++) {
142     root = root->insert(i);
143     root->print();
144     cerr << endl;
145   }
146   for (int i = 0; i < 10; i++) {
147     root = root->erase(i);
148     root->print();
149     cerr << endl;
150   }
```

```
151   return 0;
152 }
```

```cpp
#include <bits/stdc++.h>

using namespace std;

template<typename T>
class DST {
public:
  int n;
  function<T(T, T)> f;
  vector<T> v;
  vector<pair<vector<T>, vector<T>>> table[32];

  DST(vector<T> &_v, function<T(T, T)> f) : v(_v), f(f) {
    n = 1;
    while (n < v.size()) {
      n <<= 1;
    }
    v.resize(n);
    int m = n >> 1;
    for (int k = 0; m; k++, m >>= 1) {
      table[k].resize(m);
      int kshift = 1 << k;
      int piv = kshift;
      for (int i = 0; i < m; i++) {
        table[k][i].first.resize(kshift);
        table[k][i].second.resize(kshift);
        table[k][i].first[0] = v[piv - 1];
        table[k][i].second[0] = v[piv];
        for (int j = 1; j < kshift; j++) {
          table[k][i].first[j] = f(table[k][i].first[j - 1], v[piv - 1 - j]);
          table[k][i].second[j] = f(table[k][i].second[j - 1], v[piv + j]);
        }
        piv += 2 << k;
      }
    }
  }

  T query(int l, int r) {
    if (r - l <= 1)
      return v[l];
    r--;
    int k = 31 - __builtin_clz(l ^ r);
    int index = l >> (k + 1);
    int piv = index * (2 << k) + (1 << k);
    return f(table[k][index].first[piv - l - 1],
             table[k][index].second[r - piv]);
  }
};
// ref: http://noshi91.hatenablog.com/entry/2018/05/08/183946
// ref: https://discuss.codechef.com/t/tutorial-disjoint-sparse-table/17404
```

```cpp

int main() {
  int n;
  cin >> n;
  vector<int> v(n);
  for (int i = 0; i < n; i++)
    v[i] = i + 1;
  DST<int> dst(v, [](int a, int b) { return max(a, b); });
  for (int i = 0; i < n; i++) {
    for (int j = i + 1; j <= n; j++) {
      int q = dst.query(i, j);
      assert(q > 0);
    }
  }
  return 0;
}
```

```cpp
#include <bits/stdc++.h>

using namespace std;
using ll = long long;

long long gcd(long long a, long long b) {
  if (a % b == 0)
    return b;
  return gcd(b, a % b);
}

// return (x, y) (a * x + b * y = gcd(a, b))
pair<long long, long long> exeuclid(long long a, long long b) {
  if (b == 0)
    return make_pair(1, 0);
  pair<long long, long long> ret = exeuclid(b, a % b);
  ret.first -= a / b * ret.second;
  return make_pair(ret.second, ret.first);
}

// v := [(modulo, reminder)]
// moduloes should be coprime
long long CRT(vector<pair<ll, ll>> v) {
  long long mod = 1;
  long long ret = 0;
  for (int i = 0; i < v.size(); i++) {
    pair<ll, ll> xy = exeuclid(mod, v[i].first);
    long long M = mod * v[i].first;
    long long l = ((v[i].second * mod) % M) * xy.first;
    l %= M;
    long long r = ((ret * v[i].first) % M) * xy.second;
    r %= M;
    ret = (l + r) % M;
    ret += M;
    ret %= M;
    mod = M;
  }
  return ret;
}

ll mod_pow(ll e, ll x, ll p) {
  ll r = 1;
  e %= p;
  while (x) {
    if (x & 1) {
      r *= e;
      r %= p;
    }
    e *= e;
    e %= p;
```

```
  51     x >>= 1;
  52   }
  53   return r;
  54 }
  55
  56 template<class T>
  57 T extgcd(T a, T b, T &x, T &y) {
  58   for (T u = y = 1, v = x = 0; a;) {
  59     T q = b / a;
  60     swap(x -= q * u, u);
  61     swap(y -= q * v, v);
  62     swap(b -= q * a, a);
  63   }
  64   return b;
  65 }
  66
  67 ll mod_inv(ll a, ll b) {
  68   return (exeuclid(a, b).first % b + b) % b;
  69 }
  70
  71 template<int mod, int primitive_root>
  72 class NTT {
  73 public:
  74   int get_mod() const { return mod; }
  75
  76   void _ntt(vector<ll> &a, int sign) {
  77     const int n = a.size();
  78     assert((n ^ (n & -n)) == 0); //n = 2^k
  79
  80     const int g = 3; //g is primitive root of mod
  81     int h = (int) mod_pow(g, (mod - 1) / n, mod); // h^n = 1
  82     if (sign == -1) h = (int) mod_inv(h, mod); //h = h^-1 % mod
  83
  84     //bit reverse
  85     int i = 0;
  86     for (int j = 1; j < n - 1; ++j) {
  87       for (int k = n >> 1; k > (i ^= k); k >>= 1);
  88       if (j < i) swap(a[i], a[j]);
  89     }
  90
  91     for (int m = 1; m < n; m *= 2) {
  92       const int m2 = 2 * m;
  93       const ll base = mod_pow(h, n / m2, mod);
  94       ll w = 1;
  95       for (int x = 0; x < m; x++) {
  96         for (int s = x; s < n; s += m2) {
  97           ll u = a[s];
  98           ll d = a[s + m] * w % mod;
  99           a[s] = u + d;
 100           if (a[s] >= mod) a[s] -= mod;
```

```
101           a[s + m] = u - d;
102           if (a[s + m] < 0) a[s + m] += mod;
103         }
104         w = w * base % mod;
105       }
106     }
107
108     for (auto &x : a) if (x < 0) x += mod;
109   }
110
111   void ntt(vector<ll> &input) {
112     _ntt(input, 1);
113   }
114
115   void intt(vector<ll> &input) {
116     _ntt(input, -1);
117     const int n_inv = mod_inv(input.size(), mod);
118     for (auto &x : input) x = x * n_inv % mod;
119   }
120
121   // 畳み込み演算を行う
122   vector<ll> convolution(const vector<ll> &a, const vector<ll> &b) {
123     int ntt_size = 1;
124     while (ntt_size < a.size() + b.size()) ntt_size *= 2;
125
126     vector<ll> _a = a, _b = b;
127     _a.resize(ntt_size);
128     _b.resize(ntt_size);
129
130     ntt(_a);
131     ntt(_b);
132
133     for (int i = 0; i < ntt_size; i++) {
134       (_a[i] *= _b[i]) %= mod;
135     }
136
137     intt(_a);
138     return _a;
139   }
140 };
141
142 ll garner(vector<pair<ll, ll>> mr, int mod) {
143   mr.emplace_back(mod, 0);
144
145   vector<ll> coffs(mr.size(), 1);
146   vector<ll> constants(mr.size(), 0);
147   for (int i = 0; i < mr.size() - 1; i++) {
148     // coffs[i] * v + constants[i] == mr[i].second (mod mr[i].first) を解く
149     ll v = (mr[i].second - constants[i]) * mod_inv(coffs[i], mr[i].first) % mr[i].first;
150     if (v < 0) v += mr[i].first;
```

```
151
152     for (int j = i + 1; j < mr.size(); j++) {
153       (constants[j] += coffs[j] * v) %= mr[j].first;
154       (coffs[j] *= mr[i].first) %= mr[j].first;
155     }
156   }
157
158   return constants[mr.size() - 1];
159 }
160
161 typedef NTT<167772161, 3> NTT_1;
162 typedef NTT<469762049, 3> NTT_2;
163 typedef NTT<1224736769, 3> NTT_3;
164 NTT_1 ntt1;
165 NTT_2 ntt2;
166 NTT_2 ntt3;
167 // ref: https://math314.hateblo.jp/entry/2015/05/07/014908
168
169 // NTT
170 // c[i] = for j,k a[j] + b[k] if j + k == i mod P
171 vector<ll> mod_conv(vector<ll> a, vector<ll> b, ll P) {
172   for (auto &i : a) i %= P;
173   for (auto &i : b) i %= P;
174   auto v1 = ntt1.convolution(a, b);
175   auto v2 = ntt2.convolution(a, b);
176   auto v3 = ntt3.convolution(a, b);
177   int n = v2.size();
178   vector<ll> ret(n);
179   for (int i = 0; i < n; i++) {
180     if (v1[i] != v2[i]) cerr << v1[i] << " " << v2[i] << endl;
181     ret[i] = garner(vector<pair<ll, ll>>(
182         {
183           make_pair(ntt1.get_mod(), v1[i]),
184           make_pair(ntt2.get_mod(), v2[i]),
185           make_pair(ntt3.get_mod(), v3[i])
186         }), P);
187   }
188   return ret;
189 }
190
191 int main() {
192   int n;
193   cin >> n;
194   vector<ll> a(n + 1), b(n + 1);
195   for (int i = 1; i <= n; i++) {
196     cin >> a[i] >> b[i];
197   }
198   auto c = mod_conv(a, b, 1000000007);
199   for (int i = 1; i <= 2 * n; i++) {
200     cout << c[i] << endl;
```

```
201   }
202   return 0;
203 }
```

```cpp
#include <bits/stdc++.h>

using namespace std;
using ll = long long;
using ld = long double;
using cx = complex<ld>;

ld pi = 3.14159265358979323846264338327950288841971;

vector<cx> dft(vector<cx> f, int n) {
  if (n == 1) return f;
  vector<cx> f0(n / 2), f1(n / 2);
  for (int i = 0; i < n / 2 && i * 2 < f.size(); i++) f0[i] = f[i * 2];
  for (int i = 0; i < n / 2 && i * 2 + 1 < f.size(); i++) f1[i] = f[i * 2 + 1];
  auto f0c = dft(f0, n / 2);
  auto f1c = dft(f1, n / 2);
  vector<cx> fc(n);
  for (int i = 0; i < n; i++) {
    cx zeta_n_i = cx(cos(2 * pi * i / n), sin(2 * pi * i / n));
    fc[i] = f0c[i % (n / 2)] + zeta_n_i * f1c[i % (n / 2)];
  }
  return fc;
}

vector<cx> idft(vector<cx> fc, int n) {
  auto fcc = dft(fc, n);
  vector<cx> swaped(n);
  for (int i = 0; i < n; i++) swaped[i] = fcc[(n - i) % n];
  vector<cx> ret(n);
  for (int i = 0; i < n; i++) ret[i] = swaped[i] / cx(n, 0);
  return ret;
}

vector<cx> mul(vector<cx> a, vector<cx> b) {
  int n = 1;
  while (n < a.size() + b.size() + 1) {
    n <<= 1;
  }
  vector<cx> cc(n);
  auto ac = dft(a, n);
  auto bc = dft(b, n);
  for (int i = 0; i < n; i++) cc[i] = ac[i] * bc[i];
  return idft(cc, n);
}

int main() {
  int n;
  cin >> n;
  vector<cx> a(n + 1), b(n + 1);
  for (int i = 1; i <= n; i++) cin >> a[i] >> b[i];
```

```
51    auto c = mul(a, b);
52    for (int i = 1; i <= 2 * n; i++) {
53      cout << (int) round(c[i].real() + .1) << endl;
54    }
55
56    return 0;
57  }
```

```cpp

class Increase {
public:
  int n, flowMax = 1000000;
  vector <vector<int>> p;
  vector<int> parent;

  Increase(int _n, vector <vector<int>> _p) {
    n = _n;
    p = _p;
    parent.resize(n);
  }

  int dfs(int id, int from, int flow) {
    if (parent[id] != id) return 0;
    parent[id] = from;
    if (id == n - 1) return flow;
    for (int i = 0; i < n; i++)
      if (p[id][i] > 0) {
        int next = dfs(i, id, min(flow, p[id][i]));
        if (next > 0) return next;
      }
    return 0;
  }

  int solve() {
    int ret = 0, flow;
    for (int i = 0; i < n; i++) parent[i] = i;

    while ((flow = dfs(0, -1, flowMax)) > 0) {
      int now = n - 1;
      ret += flow;
      while (parent[now] >= 0) {
        p[parent[now]][now] -= flow;
        p[now][parent[now]] += flow;
        now = parent[now];
      }
      for (int i = 0; i < n; i++) parent[i] = i;
    }
    return ret;
  }
};


class Increase {
public:
  int n, flowMax = 1000000;
  vector <map<int, int>> p;
  vector<int> parent;
  vector<int> stack;
```

```
51
52    Increase(int n, vector <map<int, int>> &p) : n(n), p(p) {
53      parent.resize(n);
54    }
55
56    int dfs(int id, int from, int flow) {
57      if (parent[id] != id) return 0;
58      stack.push_back(id);
59      parent[id] = from;
60      if (id == n - 1) return flow;
61      for (auto &v : p[id])
62        if (v.second > 0) {
63          int next = dfs(v.first, id, min(flow, v.second));
64          if (next > 0) return next;
65        }
66      return 0;
67    }
68
69    int solve() {
70      int ret = 0, flow;
71      for (int i = 0; i < n; i++) parent[i] = i;
72
73      while ((flow = dfs(0, -1, flowMax)) > 0) {
74        int now = n - 1;
75        ret += flow;
76        while (parent[now] >= 0) {
77          p[parent[now]][now] -= flow;
78          p[now][parent[now]] += flow;
79          now = parent[now];
80        }
81        for (auto &v: stack) parent[v] = v;
82        stack.clear();
83      }
84      return ret;
85    }
86  };
```

```cpp
// 幾何ライブラリ 2次元ベクトル内積/外積 線分距離 直線交点 凸包 円
#include <bits/stdc++.h>

using namespace std;

using ll = long long;
using vi = vector<int>;
using vll = vector<ll>;
using vvi = vector<vector<int>>;
using vvl = vector<vector<ll>>;

double eps = 0.0000001;
// asin(1.) * 2
double pi = 3.14159265358979323846264338327950288;
using p2 = complex<double>;
// x: real
// y: imag

double det(p2 v1, p2 v2) {
  return v1.real() * v2.imag() - v1.imag() * v2.real();
}

double dot(p2 v1, p2 v2) {
  return v1.real() * v2.real() + v1.imag() * v2.imag();
}

double dist2(p2 v) {
  return dot(v, v);
}

double dist(p2 v) {
  return sqrt(dist2(v));
}

bool same(double x, double y) { return abs(x - y) < eps; }

double dist2(p2 l1, p2 l2) { return dot(l1 - l2, l1 - l2); }

double dist(p2 l1, p2 l2) {
  return sqrt(dist2(l1, l2));
}

int ccw(p2 a, p2 b, p2 c) {
  b -= a;
  c -= a;
  if (det(b, c) > eps)return 1;
  if (det(b, c) < -eps) return -1;
  if (dot(b, c) < -eps) return 2;
  if (dist2(b) < dist2(c)) return -2;
  return 0;
```

```
 51  }
 52
 53  auto p2comp = [](const p2 &l, const p2 &r) {
 54    if (abs(l.real() - r.real()) > eps)
 55      return l.real() < r.real();
 56    return l.imag() < r.imag();
 57  };
 58
 59  struct Line {
 60    p2 st, ed;
 61
 62    Line(p2 st, p2 ed) : st(st), ed(ed) {}
 63
 64    Line(double x1, double y1, double x2, double y2)
 65        : st(p2(x1, y1)), ed(p2(x2, y2)) {}
 66
 67    Line(p2 st, double x, double y) : st(st), ed(p2(x, y)) {}
 68
 69    Line(double x, double y, p2 ed) : st(p2(x, y)), ed(ed) {}
 70
 71    double dist() { return sqrt(dist2(st, ed)); }
 72
 73    bool isPalla(Line l) { return abs(det(ed - st, l.ed - l.st)) < eps; }
 74
 75    double x() { return ed.real() - st.real(); }
 76
 77    double y() { return ed.imag() - st.imag(); }
 78
 79    p2 v() { return ed - st; }
 80  };
 81
 82  // l1.st + (l1.st - l1.ed) * r.first = l2.st + (l2.st - l2.ed) * r.second
 83  // 方程式を満たす(r.first, r.second)を返す
 84  // l1.isPalla(l2) => (nan, nan)
 85  pair<double, double> interP(Line l1, Line l2) {
 86    double a = l1.x();
 87    double b = -l2.x();
 88    double c = l1.y();
 89    double d = -l2.y();
 90    double inv = 1. / (a * d - c * b);
 91    double e1 = -l1.st.real() + l2.st.real();
 92    double e2 = -l1.st.imag() + l2.st.imag();
 93    return make_pair((d * e1 - b * e2) * inv, (-c * e1 + a * e2) * inv);
 94  }
 95
 96  bool intersec(Line l1, Line l2) {
 97    if (l1.isPalla(l2))
 98      return false;
 99    auto r = interP(l1, l2);
100    return eps < r.first && r.first < 1. - eps && eps < r.second &&
```

```
101         r.second < 1. - eps;
102 }
103
104 double inter_r(Line l, p2 c) {
105   p2 a = l.st, b = l.ed;
106   return dot(a - c, l.v()) / dist2(l.v());
107 }
108
109 double dist(Line l, p2 c) {
110   double r = inter_r(l, c);
111   if (r < -eps) return dist(l.st, c);
112   if (1. + eps < r) return dist(l.ed, c);
113   return dist(l.st + l.v() * r, c);
114 }
115
116 p2 nearest(Line l, p2 c) {
117   double r = inter_r(l, c);
118   if (r < -eps) return l.st;
119   if (1. + eps < r) return l.ed;
120   return l.st + l.v() * r;
121 }
122
123 double dist(Line l1, Line l2) {
124   return min(min(dist(l1, l2.st), dist(l1, l2.ed)), min(dist(l2, l1.st), dist(l2, l1.ed)));
125 }
126
127 struct Poly {
128   vector<p2> ps;
129   double d;
130
131   Poly(vector<p2> ps) : ps(ps) {
132     d = 0;
133     for (int i = 0; i < ps.size(); i++) d += dist(ps[i], ps[(i + 1) % ps.size()]);
134   }
135
136   // 頂点上/辺上は微妙
137   bool include(p2 p) {
138     // 半直線
139     Line l(p, p2(-10000, -1));
140     int c = 0;
141     for (int i = 0; i < ps.size(); i++) {
142       if (intersec(l, Line(ps[i], ps[(i + 1) % ps.size()]))) c++;
143     }
144     return c % 2 == 1;
145   }
146
147   bool include(p2 p, bool on_vert, bool on_edge) {
148     for (auto &q : ps) if (dist(p, q) < eps) return on_vert;
149     for (int i = 0; i < ps.size(); i++) {
150       if (ccw(ps[i], ps[(i + 1) % ps.size()], p) == 0) return on_edge;
```

```
151     }
152     return include(p);
153   }
154
155   bool intersecl(Line l) {
156     for (int i = 0; i < ps.size(); i++) {
157       if (intersec(l, Line(ps[i], ps[(i + 1) % ps.size()])))
158         return true;
159     }
160     return false;
161   }
162 };
163
164 struct Circle {
165   p2 p;
166   double r;
167
168   Circle(p2 p, double r) : p(p), r(r) {}
169
170   bool include(p2 l) { return dist2(p, l) < r * r + eps; }
171
172   // 円同士の交点
173   // 存在すれば2つ
174   vector<p2> intersec(Circle c) {
175     p2 diff = c.p - p;
176     double dist = dot(diff, diff);
177     double a = (dist + r * r - c.r * c.r) / 2.;
178     double D = dist * r * r - a * a;
179     if (D < eps)
180       return vector<p2>();
181     double Dsqrt = sqrt(D);
182     vector<p2> ps;
183     ps.emplace_back((a * diff.real() + diff.imag() * Dsqrt) / dist + p.real(),
184                     (a * diff.imag() - diff.real() * Dsqrt) / dist + p.imag());
185     ps.emplace_back((a * diff.real() - diff.imag() * Dsqrt) / dist + p.real(),
186                     (a * diff.imag() + diff.real() * Dsqrt) / dist + p.imag());
187     return ps;
188   }
189 };
190
191 // 半時計回り
192 struct ConX {
193   vector<p2> ps;
194
195   // graham scan
196   // ref: プログラミングコンテストチャレンジブック p233
197   ConX(vector<p2> v) {
198     sort(v.begin(), v.end(), p2comp);
199
200     int k = 0, n = v.size();
```

```
201     ps.resize(n * 2);
202     for (int i = 0; i < n; i++) {
203       while (k > 1 && det(ps[k - 1] - ps[k - 2], v[i] - ps[k - 1]) < eps)
204         k--;
205       ps[k++] = v[i];
206     }
207     for (int i = n - 2, t = k; i >= 0; i--) {
208       while (k > t && det(ps[k - 1] - ps[k - 2], v[i] - ps[k - 1]) < eps)
209         k--;
210       ps[k++] = v[i];
211     }
212     ps.resize(k - 1);
213   }
214
215   Poly toPoly() {
216     return Poly(ps);
217   }
218
219   size_t size() { return ps.size(); }
220 };
221
222 int n;
223 vector<double> rs;
224 vector<p2> ps;
225 vector<Circle> cs;
226
227 bool f(double l) {
228   cs.clear();
229   for (int i = 0; i < n; i++) {
230     double rr = rs[i] * rs[i] - l * l;
231     if (rr < eps)
232       return false;
233     cs.emplace_back(ps[i], sqrt(rr));
234   }
235   vector<p2> may;
236   for (int i = 0; i < n; i++) {
237     may.push_back(cs[i].p);
238     for (int j = i + 1; j < n; j++) {
239       auto v = cs[i].intersec(cs[j]);
240       if (v.size() == 0)
241         continue;
242       may.push_back(v[0]);
243       may.push_back(v[1]);
244     }
245   }
246   for (auto &p : may) {
247     bool ok = true;
248     for (auto &c : cs) {
249       if (!c.include(p)) {
250         ok = false;
```

```
251         break;
252       }
253     }
254     if (ok)
255       return true;
256   }
257   return false;
258 }
259
260 int main() {
261   printf("%.20lf\n", asin(1.) * 2);
262   cin.tie(nullptr);
263   ios::sync_with_stdio(false);
264   Poly p(vector<p2>({p2(0, 0), p2(1, 0.5), p2(2, 0), p2(2, 2), p2(1, 1.5), p2(0, 2)})
      );
265   vector<double> xy({-1, 0, 1, 2, 3});
266   for (auto &x : xy)
267     for (auto &y : xy) {
268       cerr << x << " " << y << " " << (p.include(p2(x, y), true, true) ? "YES" : "NO")
      << endl;
269     }
270
271   return 0;
272 }
273
```

```cpp
#include <bits/stdc++.h>

using namespace std;

class G {
public:
  vector<vector<int>> p, inv, sp, sinv;
  int n, m;
  vector<int> b, c, depth;
  vector<bool> a;

  G(int nn, vector<vector<int>> &np, vector<vector<int>> &ninv) {
    n = nn;
    p = np;
    inv = ninv;
    a.resize(n);
    c.resize(n);
    for (int i = 0; i < n; i++) {
      c[i] = -1;
    }
  }

  void dfs(int i) {
    if (a[i])
      return;
    a[i] = true;
    for (int j = 0; j < p[i].size(); j++) {
      if (a[p[i][j]]) {
        continue;
      }
      dfs(p[i][j]);
    }
    b.push_back(i);
  }

  void dfs2(int i, int id) {
    if (c[i] > -1)
      return;
    c[i] = id;
    for (int j = 0; j < inv[i].size(); j++) {
      if (c[inv[i][j]] > -1)
        continue;
      dfs2(inv[i][j], id);
    }
  }

  void solve() {
    for (int i = 0; i < n; i++) {
      dfs(i);
    }
```

```
51      int j = 0;
52      for (int i = n - 1; i >= 0; i--) {
53        if (c[b[i]] > -1)
54          continue;
55        dfs2(b[i], j++);
56      }
57      m = j;
58      sp.resize(m);
59      sinv.resize(m);
60      for (int i = 0; i < n; i++) {
61        for (int j = 0; j < p[i].size(); j++) {
62          if (c[i] == c[p[i][j]])
63            continue;
64          sp[c[i]].push_back(c[p[i][j]]);
65          sinv[c[p[i][j]]].push_back(c[i]);
66        }
67      }
68    }
69  };
70
71  struct SAT {
72    int n;
73    vector<pair<int, int>> conjs;
74
75    SAT() {
76      n = 0;
77    }
78
79    // a \/ b
80    void add_or(int a, int b) {
81      add_impl(-a, b);
82      add_impl(-b, a);
83    }
84
85    // a /\ b
86    void add_and(int a, int b) {
87      add_impl(a, a);
88      add_impl(b, b);
89    }
90
91    // a
92    void add_lit(int a) {
93      add_impl(a, a);
94    }
95
96    // a ^ b
97    void add_xor(int a, int b) {
98      add_or(a, b);
99      add_or(-a, -b);
100   }
```

```
101
102   // a => b
103   void add_impl(int a, int b) {
104     n = max(n, max(abs(a), abs(b)));
105     conjs.emplace_back(a, b);
106   }
107
108   // not(a \/ b)
109   void add_nor(int a, int b) {
110     add_and(-a, -b);
111   }
112
113   int f(int a) {
114     assert(a != 0);
115     if (a > 0) return a - 1;
116     else return n + abs(a) - 1;
117   }
118
119   bool solve() {
120     vector<vector<int>> p(2 * n), inv(2 * n);
121     for (auto &c : conjs) {
122       p[f(c.first)].push_back(f(c.second));
123       inv[f(c.second)].push_back(f(c.first));
124     }
125
126     G g(2 * n, p, inv);
127     g.solve();
128     bool ok = true;
129     for (int i = 0; i < n; i++) if (g.c[i] == g.c[i + n]) ok = false;
130     return ok;
131   }
132 };
133
134 int main() {
135   int n, m;
136   cin >> n >> m;
137   vector<int> l(n), r(n);
138   for (int i = 0; i < n; i++) {
139     cin >> l[i] >> r[i];
140     r[i]++;
141   }
142   SAT solver;
143
144   // p_1, .., p_2n : 左右
145   vector<int> mp(2 * n);
146   for (int i = 0; i < n; i++) {
147     mp[2 * i] = i + 1;
148     mp[2 * i + 1] = -(i + 1);
149   }
150   vector<pair<int, int>> ps(2 * n);
```

```
151   for (int i = 0; i < n; i++) {
152     ps[i * 2] = make_pair(l[i], r[i]);
153     ps[i * 2 + 1] = make_pair(m - r[i], m - l[i]);
154   }
155   for (int i = 0; i < n * 2; i++) {
156     for (int j = i + 1; j < 2 * n; j++) {
157       if (i / 2 == j / 2) continue;
158       if (!(ps[i].second <= ps[j].first || ps[j].second <= ps[i].first)) {
159         solver.add_or(-mp[i], -mp[j]);
160       }
161     }
162   }
163
164   if (solver.solve()) {
165     cout << "YES" << endl;
166   } else {
167     cout << "NO" << endl;
168   }
169
170   return 0;
171 }
172
```

```cpp

class Bucket {
public:
  int size;
  vector<long long> v;
  long long addSum;
  long long stageNum;

  Bucket(int n) {
    size = n;
    v.resize(n);
    addSum = 0;
    stageNum = 0;
  }

  // add a [l, r)
  void add(int l, int r, long long a) {
    if (l <= 0 && r >= size) {
      addSum += a;
    } else {
      for (int i = max(0, l); i < min(size, r); i++)
        v[i] += a;
    }
  }

  // add [l, r)
  int addStage(int l, int r, long long a) {
    if (l <= 0 && r >= size) {
      stageNum++;
      add(l, r, a);
      return size;
    } else {
      int c = 0;
      for (int i = max(0, l); i < min(size, r); i++) {
        v[i] += a + c++;
      }
      return c;
    }
  }

  long long get(int i) {
    if (stageNum > 0) {
      for (int i = 0; i < size; i++)
        v[i] += stageNum * i;
      stageNum = 0;
    }
    if (addSum > 0) {
      for (int i = 0; i < size; i++)
        v[i] += addSum;
      addSum = 0;
```

```
51      }
52      return v[i];
53    }
54  };
55
56  const int BSIZE = 300;
57
58  class Buckets {
59  public:
60    vector <Bucket> b;
61    int size;
62    int bsize;
63    int num;
64
65    Buckets(int n) {
66      size = n;
67      bsize = (n + BSIZE - 1) / BSIZE;
68      for (; n > 0; n -= bsize) {
69        Bucket c(min(n, bsize));
70        b.push_back(c);
71      }
72      num = b.size();
73    }
74
75    void add(int l, int r, long long a) {
76      for (int i = 0; i < num; i++)
77        b[i].add(l - bsize * i, r - bsize * i, a);
78    }
79
80    int addStage(int l, int r) {
81      int c = 0;
82      for (int i = 0; i < num; i++) {
83        c += b[i].addStage(l - bsize * i, r - bsize * i, c);
84      }
85      return c;
86    }
87
88    long long get(int i) { return b[i / bsize].get(i - (i / bsize) * bsize); }
89  };
90
```

```cpp
#include <bits/stdc++.h>

using namespace std;

typedef long long ll;
typedef vector<int> vi;
typedef vector<ll> vll;
typedef vector<vector<int>> vvi;
typedef vector<vector<ll>> vvl;

// maximum independent set
// O(2^n * n) : n <= 20
// O(2^n/2 * n) : otherwise
//
// param v
// v[i] := NG edge (not contain self)
struct I {
  int n;
  vll ng;

  I(vll &v) {
    n = v.size();
    ng = v;
  }

  int solve() {
    if (n <= 20) {
      int size = 1 << n;
      int ret = 0;
      for (int i = 0; i < size; i++) {
        int out = 0;
        int size = 0;
        for (int j = 0; j < n; j++) {
          out |= ((i >> j) & 1) ? ng[j] : 0;
          size += ((i >> j) & 1);
        }
        ret = max(ret, (i & out) ? 0 : size);
      }
      return ret;
    } else {
      int m = n - 20;
      int size = 1 << m;
      vi dp(size);
      dp[0] = 0;
      for (int i = 0; i < m; i++)
        dp[1 << i] = 1;
      for (int i = 1; i < size; i++) {
        if (!((i - 1) & i))
          continue;
        int out = 0;
```

```
51        int size = 0;
52        dp[i] = 0;
53        for (int j = 0; j < m; j++) {
54          dp[i] = max(dp[i], ((i >> j) & 1) ? dp[i ^ (1 << j)] : 0);
55        }
56        for (int j = 0; j < m; j++) {
57          out |= ((i >> j) & 1) ? ng[j] : 0;
58          size += ((i >> j) & 1);
59        }
60        dp[i] = max(dp[i], (i & out) ? 0 : size);
61      }
62      size = 1 << 20;
63      int ret = 0;
64      int mask = (1 << m) - 1;
65      for (int i = 0; i < size; i++) {
66        ll out = 0;
67        int size = 0;
68        for (int j = m; j < n; j++) {
69          out |= ((i >> (j - m)) & 1) ? ng[j] : 0;
70          size += ((i >> (j - m)) & 1);
71        }
72        ret = max(ret, ((out >> m) & i) ? 0 : size + dp[(mask & out) ^ mask]);
73      }
74      return ret;
75    }
76  }
77 };
78
79 int main() {
80   cin.tie(0);
81   ios::sync_with_stdio(false);
82   vll v(21);
83   for (int i = 0; i < 21; i++) {
84     int s = (1 << 22) - 1;
85     v[i] = s ^ (1 << i);
86   }
87   I i(v);
88   cout << i.solve() << endl;
89   return 0;
90 }
91
```

```cpp
#include <algorithm>
#include <fstream>
#include <iomanip>
#include <iostream>
#include <map>
#include <math.h>
#include <set>
#include <stdio.h>
#include <string>
#include <utility>
#include <vector>

using namespace std;

using ll = long long;

const long long P = 1000000007;

class C {
public:
  int n;
  vector<long long> fac, inv, facInv;

  long long power(long long e, long long x) {
    if (x == 0)
      return 1;
    if (x == 1)
      return e;
    if (x % 2 == 0)
      return power((e * e) % P, x / 2);
    return (e * power(e, x - 1)) % P;
  }

  C(int n_) {
    n = n_;
    fac.resize(n + 1);
    inv.resize(n + 1);
    facInv.resize(n + 1);
    fac[0] = fac[1] = 1;
    for (int i = 2; i <= n; i++)
      fac[i] = (i * fac[i - 1]) % P;
    inv[0] = inv[1] = 1;
    for (int i = 2; i <= n; i++)
      inv[i] = power(i, P - 2);
    facInv[0] = facInv[1] = 1;
    for (int i = 2; i <= n; i++)
      facInv[i] = (inv[i] * facInv[i - 1]) % P;
  }

  long long comb(int N, int K) {
```

```
51      if (N < K || K < 0 || N < 0)
52        return 0;
53      if (N == 0 || K == 0 || K == N)
54        return 1;
55      return ((fac[N] * facInv[K]) % P * facInv[N - K]) % P;
56    }
57
58    long long hcomb(int N, int K) {
59      if (N == 0 && K == 0)
60        return 1;
61      return comb(N + K - 1, K);
62    }
63
64    long long mul(long long a, long long b) { return (a * b) % P; }
65
66    long long add(long long a, long long b) { return (a + b) % P; }
67
68    // find c
69    // where a^c = b (mod P)
70    long long dlp(long long a, long long b) {
71      long long m = ceil(sqrt(P));
72      map<long long, long long> mp;
73      for (int i = 0; i < m; i++)
74        mp[power(a, i)] = i;
75      long long ainvm = power(power(a, m), P - 2);
76      cout << m << endl;
77      for (int i = 0; i < m; i++) {
78        if (mp.find(b) != mp.end()) {
79          return i * m + mp[b];
80        }
81        b = mul(b, ainvm);
82      }
83      return -1;
84    }
85  };
86
87  template<long long M>
88  class mint {
89  public:
90    ll a;
91
92    mint(ll a = 0) : a(a % M) {}
93
94    ll &v() { return a; }
95
96    ll const &v() const { return a; }
97
98    mint operator+(const mint rhs) const {
99      return mint(*this) += rhs;
100   }
```

```
101
102   mint operator-(const mint rhs) const {
103     return mint(*this) -= rhs;
104   }
105
106   mint operator*(const mint rhs) const {
107     return mint(*this) *= rhs;
108   }
109
110   mint operator/(const mint rhs) const {
111     return mint(*this) /= rhs;
112   }
113
114   mint pow(ll x) const {
115     mint ret(1);
116     mint acc = a;
117     while (x > 0) {
118       if (x % 2) {
119         ret *= acc;
120       }
121       acc *= acc;
122       x >>= 1;
123     }
124     return ret;
125   }
126
127   mint &operator+=(const mint rhs) {
128     a += rhs.a;
129     a %= M;
130     return *this;
131   }
132
133   mint &operator-=(const mint rhs) {
134     a -= rhs.a + M;
135     a %= M;
136     return *this;
137   }
138
139   mint &operator*=(const mint rhs) {
140     a *= rhs.a;
141     a %= M;
142     return *this;
143   }
144
145   mint &operator/=(const mint rhs) {
146     a *= rhs.pow(M - 2).v();
147     a %= M;
148     return *this;
149   }
150
```

```
151 };
152
153 std::ostream &operator<<(ostream &stream, const mint<P> &m) {
154   return stream << m.v();
155 }
156
157
158 ll power(ll e, ll x) {
159   if (x == 0) return 1;
160   ll acc = e;
161   ll ret = 1;
162   while (x > 0) {
163     if (x % 2) {
164       ret *= acc;
165     }
166     acc *= acc;
167     x >>= 1;
168   }
169   return ret;
170 }
171
172 ll tot(ll x) {
173   map<ll, int> primes;
174   ll y = x;
175   for (ll i = 2; i * i <= y; i++) {
176     if (y % i == 0) {
177       primes[i]++;
178       y /= i;
179       i--;
180     }
181   }
182   if (y > 1) primes[y]++;
183
184   ll ret = 1;
185   for (auto &it : primes) {
186     ret *= (power(it.first, it.second) - power(it.first, it.second - 1));
187   }
188   return ret;
189 }
190
191 int main() {
192   C cmp(100);
193   long long c = cmp.dlp(100, 192971657);
194   cout << c << endl;
195   cout << cmp.power(100, c) << endl;
196   cout << cmp.power(100, 10000000) << endl;
197
198   mint<P> a(10), b(100), d(100);
199   cout << d / a << endl;
200   cout << tot(1000000007) << endl;
```

```
201
202   return 0;
203 }
204
```

```cpp
#include <bits/stdc++.h>

using namespace std;

using ll = long long;
using vi = vector<int>;
using vll = vector<ll>;
using vvi = vector<vector<int>>;
using vvl = vector<vector<ll>>;

const int MAX_STRING_SIZE = 1000000;

ll base = 1007;

vector<ll> MODS({2147483647, 6700417, 524287, 1000000007}

);

vector<vector<ll>> invs(MODS.size()), pows(MODS.size());

ll powm(ll e, ll x, ll m) {
  if (x == 0) return 1;
  if (x == 1) return e;
  if (x % 2 == 0) return powm((e * e) % m, x / 2, m);
  return (e * powm(e, x - 1, m)) % m;
}

void RHinit() {
  int i = 0;
  for (auto &p : MODS) {
    invs[i].resize(MAX_STRING_SIZE + 1);
    invs[i][0] = 1;
    ll inv = powm(base, p - 2, p);
    for (int j = 1; j <= MAX_STRING_SIZE; j++) {
      invs[i][j] = (invs[i][j - 1] * inv) % p;
    }
    pows[i].resize(MAX_STRING_SIZE + 1);
    pows[i][0] = 1;
    for (int j = 1; j <= MAX_STRING_SIZE; j++) {
      pows[i][j] = (pows[i][j - 1] * base) % p;
    }
    i++;
  }
}

struct RH {
  int n;
  string s;
  vector<vector<ll>> hs;

```

```
 51   RH(string &s) : s(s) {
 52     if (pows[0].empty()) {
 53       RHinit();
 54     }
 55     n = s.size();
 56     hs.resize(MODS.size());
 57     for (int i = 0; i < MODS.size(); i++) {
 58       hs[i].resize(n + 1);
 59       hs[i][0] = 0;
 60       for (int j = 0; j < n; j++) {
 61         hs[i][j + 1] = (hs[i][j] + pows[i][j] * s[j]) % MODS[i];
 62       }
 63     }
 64   }
 65
 66   vector<ll> substr(int l, int r) {
 67     vector<ll> ret(MODS.size());
 68     for (int i = 0; i < MODS.size(); i++) {
 69       ret[i] = (hs[i][r] - hs[i][l] + MODS[i]) % MODS[i];
 70       ret[i] *= invs[i][l];
 71       ret[i] %= MODS[i];
 72     }
 73     return ret;
 74   }
 75 };
 76
 77 struct N {
 78   vector<ll> v;
 79 };
 80
 81 bool operator<(const N &l, const N &r) {
 82   if (l.v[0] == r.v[0]) {
 83     if (l.v[1] == r.v[1]) {
 84       if (l.v[2] == r.v[2]) {
 85         return l.v[3] < r.v[3];
 86       }
 87       return l.v[2] < r.v[2];
 88     }
 89     return l.v[1] < r.v[1];
 90   }
 91   return l.v[0] < r.v[0];
 92 }
 93
 94 int main() {
 95   int n, m;
 96   cin >> n >> m;
 97   string s;
 98   cin >> s;
 99   RH h(s);
100   int l = 0, r = 1;
```

```
101   set<N> ss;
102   for (int i = 0; i < m; i++) {
103     string t;
104     cin >> t;
105     if (t == "L++") {
106       l++;
107       r;
108     } else if (t == "L--") {
109       l--;
110       r;
111     } else if (t == "R++") {
112       l;
113       r++;
114     } else if (t == "R--") {
115       l;
116       r--;
117     }
118     ss.insert(N{h.substr(l, r)});
119   }
120   cout << ss.size() << endl;
121   return 0;
122 }
123
```

```cpp
#include <bits/stdc++.h>

using namespace std;
using ll = long long;
using vi = vector<int>;
using vll = vector<ll>;
using vvi = vector<vector<int>>;
using vvl = vector<vector<ll>>;

class RMQ {
  vector<long long> data;

public:
  static const long long INF = 100000000000;
  int n;

  RMQ(int _) {
    n = _;
    data.resize(n * 4);
    for (int i = 0; i < n * 4; i++)
      data[i] = INF;
  }

  void update(int index, long long val) {
    int i = index + n - 1;
    data[i] = val;
    while (i > 0) {
      i = (i - 1) / 2;
      data[i] = min(data[i * 2 + 1], data[i * 2 + 2]);
    }
  }

  // [a, b)
  long long query(int a, int b, int k, int l, int r) {
    if (a < 0)
      return 0;
    if (r <= a || b <= l)
      return INF;
    if (a <= l && r <= b)
      return data[k];
    else
      return min(query(a, b, k * 2 + 1, l, (l + r) / 2),
                 query(a, b, k * 2 + 2, (r + l) / 2, r));
  }

  long long query(int a, int b) { return query(a, b, 0, 0, n); }
};

template<typename T>
class tRMQ {
```

```
51    vector<T> data;
52    T unit;
53
54  public:
55    int n;
56    function<T(const T &, const T &)> f;
57
58    tRMQ(int _, T u, function<T(T, T)> bi) {
59      unit = u;
60      f = bi;
61      n = 1;
62      while (n < _) {
63        n <<= 1;
64      }
65      data.resize(n * 4);
66      for (int i = 0; i < n * 4; i++)
67        data[i] = unit;
68    }
69
70    tRMQ(vector<T> &v, T u, function<T(T, T)> bi) {
71      unit = u;
72      f = bi;
73      n = 1;
74      while (n < v.size())
75        n <<= 1;
76      data.resize(n * 4, u);
77      for (int i = 0; i < v.size(); i++) {
78        data[n + i - 1] = v[i];
79      }
80      for (int i = n - 2; i >= 0; i--) {
81        data[i] = f(data[i * 2 + 1], data[i * 2 + 2]);
82      }
83    }
84
85    void update(int index, T val) {
86      int i = index + n - 1;
87      data[i] = val;
88      while (i > 0) {
89        i = (i - 1) / 2;
90        data[i] = f(data[i * 2 + 1], data[i * 2 + 2]);
91      }
92    }
93
94    // [a, b)
95    T query(int a, int b, int k, int l, int r) {
96      if (a < 0 || r <= a || b <= l)
97        return unit;
98      if (a <= l && r <= b)
99        return data[k];
100     else
```

```
101       return f(query(a, b, k * 2 + 1, l, (l + r) / 2),
102                query(a, b, k * 2 + 2, (r + l) / 2, r));
103     }
104
105     T query(int a, int b) { return query(a, b, 0, 0, n); }
106   };
107
108   tRMQ<ll> minrmq(int n) {
109     return tRMQ<ll>(n, 10000000000000000LL, [](ll r, ll l) { return min(l, r); });
110   }
111
112   tRMQ<ll> maxrmq(int n) {
113     return tRMQ<ll>(n, -10000000000000000LL,
114                     [](ll r, ll l) { return max(l, r); });
115   }
116
117   tRMQ<ll> sumrmq(int n) {
118     return tRMQ<ll>(n, 0, [](ll l, ll r) { return l + r; });
119   }
120
121   template<typename T>
122   struct RSQRAQ2 {
123     int n;
124     T unit;
125     function<T(T, T)> update_f, sum_f, query_f;
126     vector<T> dat, lazy;
127
128     RSQRAQ2() {}
129
130     RSQRAQ2(int n_, T unit, function<T(T, T)> update_f, function<T(T, int)> sum_f,
      function<T(T, T)> query_f)
131         : unit(unit), update_f(update_f), sum_f(sum_f), query_f(query_f) {
132       n = 1;
133       while (n < n_)
134         n *= 2;
135       dat.assign(n * 2, unit);
136       lazy.assign(n * 2, unit);
137     }
138
139     RSQRAQ2(vector<T> v, T unit, function<T(T, T)> update_f, function<T(T, int)>
      sum_f, function<T(T, T)> query_f)
140         : unit(unit), update_f(update_f), sum_f(sum_f), query_f(query_f) {
141       n = 1;
142       while (n < v.size()) n <<= 1;
143       dat.assign(n * 2, unit);
144       lazy.assign(n * 2, unit);
145       for (int i = 0; i < v.size(); i++) {
146         dat[n + i - 1] = v[i];
147       }
148       for (int i = n - 2; i >= 0; i--) {
```

```
149       dat[i] = query_f(dat[i * 2 + 1], dat[i * 2 + 2]);
150     }
151   }
152
153   void eval(int len, int k) {
154     if (lazy[k] == unit)
155       return;
156     if (k * 2 + 1 < n * 2 - 1) {
157       lazy[2 * k + 1] = update_f(lazy[2 * k + 1], lazy[k]);
158       lazy[2 * k + 2] = update_f(lazy[2 * k + 2], lazy[k]);
159     }
160     dat[k] = update_f(dat[k], sum_f(lazy[k], len));
161     lazy[k] = unit;
162   }
163
164   // [a, b)
165   T update(int a, int b, T x, int k, int l, int r) {
166     eval(r - l, k);
167     if (b <= l || r <= a)
168       return dat[k];
169     if (a <= l && r <= b) {
170       lazy[k] = update_f(lazy[k], x);
171       return query_f(dat[k], sum_f(lazy[k], r - l));
172     }
173     return dat[k] = query_f(update(a, b, x, 2 * k + 1, l, (l + r) / 2),
174                             update(a, b, x, 2 * k + 2, (l + r) / 2, r));
175   }
176
177   T update(int a, int b, T x) { return update(a, b, x, 0, 0, n); }
178
179   // [a, b)
180   T query(int a, int b, int k, int l, int r) {
181     eval(r - l, k);
182     if (b <= l || r <= a)
183       return unit;
184     if (a <= l && r <= b)
185       return dat[k];
186     return query_f(query(a, b, 2 * k + 1, l, (l + r) / 2), query(a, b, 2 * k + 2, (l + r) / 2,
      r));
187   }
188
189   T query(int a, int b) { return query(a, b, 0, 0, n); }
190 };
191 // ref:https://www65.atwiki.jp/kyopro-lib/pages/15.html
192
193 int main() {
194   vector<ll> v({1, 2, 3, 4, 5, 6, 7, 8});
195   RSQRAQ2<ll> q(v, 0, [](ll l, ll r) {
196                 return l + r;
197               }, [](ll l, int len) {
```

```
198                 return l;
199               }, [](ll l, ll r) {
200                 return max(l, r);
201               }
202     );
203     for (int i = 0; i < v.size(); i++) cerr << q.query(i, i + 1) << endl;
204     q.update(0, 8, 3);
205     cerr << q.query(0, 8) << endl;
206     for (int i = 0; i < v.size(); i++) cerr << q.query(i, i + 1) << endl;
207
208     return 0;
209   }
```

```cpp
#include <bits/stdc++.h>

using namespace std;

using ll = long long;

const int TRIE_SIZE = 26;
const int TRIE_OFFSET = 'a';

struct Trie {
  int d, size;
  Trie *child[TRIE_SIZE];

  ll score = 0;

  Trie() {
    d = 0;
    size = 0;
    for (int i = 0; i < TRIE_SIZE; i++) child[i] = nullptr;
  }

  void update(string &s, int i, ll v) {
    size++;
    d = i;
    if (i < s.size()) {
      if (child[s[i] - TRIE_OFFSET] == nullptr)
        child[s[i] - TRIE_OFFSET] = new Trie();

      child[s[i] - TRIE_OFFSET]->update(s, i + 1, v);
    } else {
      score = v;
    }
  }

  Trie *next(char c) {
    return child[c - TRIE_OFFSET];
  }

  ~Trie() {
    for (int i = 0; i < TRIE_SIZE; i++) {
      if (child[i] != nullptr) delete child[i];
    }
  }
};
```

```cpp
#include <iostream>
#include <vector>

using namespace std;

class U {
public:
  int n;
  vector<int> p, s;

  U(int _) {
    n = _;
    p.resize(n);
    s.resize(n);
    for (int i = 0; i < n; i++) {
      p[i] = i;
      s[i] = 1;
    }
  }

  bool connect(int a, int b) {
    int ap, bp;
    for (ap = p[a]; ap != p[ap]; ap = p[ap]);
    for (bp = p[b]; bp != p[bp]; bp = p[bp]);
    if (ap == bp)
      return true;
    int mi = min(ap, bp), ma = max(ap, bp);
    p[ma] = mi;
    s[mi] += s[ma];
    s[ma] = 0;
    for (int pp = a; pp != mi;) {
      int next = p[pp];
      p[pp] = mi;
      pp = next;
    }
    for (int pp = b; pp != mi;) {
      int next = p[pp];
      p[pp] = mi;
      pp = next;
    }
    return false;
  }

  int q(int a) {
    int ap;
    for (ap = a; ap != p[ap]; ap = p[ap]);
    return s[ap];
  }

  int parent(int a) {
```

```
51      int ap;
52      for (ap = a; ap != p[ap]; ap = p[ap]);
53      return p[ap];
54    }
55
56    bool query(int a, int b) { return parent(a) == parent(b); }
57  };
58
59  int main() {
60    int n;
61    cin >> n;
62    U u(n);
63  }
64
```

```cpp
#include <algorithm>
#include <fstream>
#include <iomanip>
#include <iostream>
#include <map>
#include <math.h>
#include <set>
#include <stdio.h>
#include <string>
#include <utility>
#include <vector>

using namespace std;

struct Z {
  long long val;

  void f(Z a) { val += a.val; }
};

// v'[i] = Σ v[j] (j in i)
// v = v'
void zeta(int n, vector<Z> &v) {
  for (int i = 0; i < n; i++)
    for (int j = 0; j < (1 << n); j++)
      if (j & (1 << i))
        v[j].f(v[j ^ (1 << i)]);
}
```