Financial University under the Government of Russian Federation

Viktor Chagai

Semyon Babich, Damir Bekirov, Yury Korobkov

Northern Eurasia Finals 2024

December 15, 2024

1 Data Structures 1

2 Mathematics 2

Data Structures (1)

fenwick.hh

Description: Fenwick my twizz=)

22 lines

```
struct FT {
  vector<ll> s;
  FT(int n) : s(n) {}
  void update(int pos, ll dif) { // a[pos] += dif
    for (; pos < sz(s); pos |= pos + 1) s[pos] += dif;
  }
  ll query(int pos) { // sum of values in [0, pos)
    ll res = 0;
    for (; pos > 0; pos &= pos - 1) res += s[pos-1];
    return res;
  }
  int lower_bound(ll sum) {// min pos st sum of [0, pos] >= sum
    // Returns n if no sum is >= sum, or -1 if empty sum is.
    if (sum <= 0) return -1;
    int pos = 0;
    for (int pw = 1 << 25; pw; pw >>= 1) {
      if (pos + pw <= sz(s) && s[pos + pw-1] < sum)
        pos += pw, sum -= s[pos-1];
    }
    return pos;
  }
};
```

segmentTree.hh

Description: Na otrezke mozhno gryaz delat

58 lines

```
template <typename T>
struct SegmentTree{
public:
    SegmentTree(vector <T>& _a) : n(_a.size()), a(_a) : {
        t.assign(4 * n, 0);
        mod.assign(4 * n, 0);
        build(1, 0, n);
    }
    void update(int v = 1, int tl = 0, int tr = n, int l, int r
        , T x){
        if (l >= r || tl >= tr) return;
        if (l == tl && r == tr){
            apply(v, l, r, x);
        }
        else {
            push(v, tl, tr);
            int mid = (tl + tr) >> 1;
            update(2 * v, tl, mid, l, min(r, mid), x);
            update(2 * v + 1, mid, tr, max(l, mid), r, x);
            t[v] = min(t[2 * v], t[2 * v + 1]);
        }
    }
    ll get_min(int v = 1, int tl = 0, int tr = n, int l, int r)
        {
        if (l >= r || tl >= tr) return INFLL;
        if (l == tl && r == tr){
            return t[v];
        }
        else {
            push(v, tl, tr);
            int mid = (tl + tr) >> 1;
            return min(get_min(2 * v, tl, mid, l, min(r, mid)),
                 get_min(2 * v + 1, mid, tr, max(l, mid), r));
        }
    }
private:
    int n;
    vector <T> &a;
    vector <T> t, mod;
    void build(int v, int l, int r){
        if (l == r - 1) {
            t[v] = a[l];
        }
        else {
            int mid = (l + r) >> 1;
            build(2 * v, l, mid);
            build(2 * v + 1, mid, r);
            t[v] = min(t[2 * v], t[2 * v + 1]);
        }
    }
    void apply (int v, int l, int r, ll x){
        t[v] += x;
        mod[v] += x;
    }
    void push (int v, int l, int r){
        int mid = (l + r) >> 1;
        apply(2*v, l, mid, mod[v]);
        apply(2*v + 1, mid, r, mod[v]);
        mod[v] = 0;
    }
};
```

sparseTable.hh

Description: Geek from Tyumen Region thinks that he is RMQ Data Structre

27 lines

```
template <typename T>
struct SparseTable{
public:
    SparseTable (vector <T> &_a) : n(_a.size()), a(_a) {
        init(n);
    }
    T rmq(T l, T r) {
        T t = __lg(r - l);
        return min(g[t][l], g[t][r - (1 << t)]);
    }
private:
    int n;
    vector <T> &a;
    vector <vector <T>> g;
    void init(int n) {
        int logn = __lg(n);
        g.assign(logn + 1, vector <T>(n));
        for (int i = 0; i < n; ++i) {
            g[0][i] = a[i];
        }
        for (int l = 0; l <= logn - 1; l++) {
            for (int i = 0; i + (2 << l) <= n; i++) {
                g[l + 1][i] = min(g[l][i], g[l][i + (1 << l)]);
            }
        }
    }
};
```

treap.hh

Description: Just treap=)

55 lines

```
struct Node {
  Node *l = 0, *r = 0;
  int val, y, c = 1;
  Node(int val) : val(val), y(rand()) {}
  void recalc();
};

int cnt(Node* n) { return n ? n->c : 0; }
void Node::recalc() { c = cnt(l) + cnt(r) + 1; }

template<class F> void each(Node* n, F f) {
  if (n) { each(n->l, f); f(n->val); each(n->r, f); }
}

pair<Node*, Node*> split(Node* n, int k) {
  if (!n) return {};
  if (cnt(n->l) >= k) { // "n->val >= k" for lower_bound(k)
    auto pa = split(n->l, k);
    n->l = pa.second;
    n->recalc();
    return {pa.first, n};
  } else {
    auto pa = split(n->r, k - cnt(n->l) - 1); // and just "k"
    n->r = pa.first;
    n->recalc();
    return {n, pa.second};
  }
}

Node* merge(Node* l, Node* r) {
  if (!l) return r;
  if (!r) return l;
  if (l->y > r->y) {
    l->r = merge(l->r, r);
    l->recalc();
    return l;
  } else {
    r->l = merge(l, r->l);
    r->recalc();
    return r;
  }
}

Node* ins(Node* t, Node* n, int pos) {
  auto [l,r] = split(t, pos);
  return merge(merge(l, n), r);
}

// Example application: move the range [l, r) to index k
void move(Node*& t, int l, int r, int k) {
  Node *a, *b, *c;
  tie(a,b) = split(t, l); tie(b,c) = split(b, r - l);
  if (k <= l) t = merge(ins(a, b, k), c);
  else t = merge(a, ins(c, b, k - r));
}
```

lazySegmentTree.hh

Description: Segment tree with ability to add or set values of large intervals, and compute max of intervals.
Usage: `Node* tr = new Node(v, 0, sz(v));`
Time: $\mathcal{O}(\log N)$.

50 lines

```
const int inf = 1e9;
struct Node {
  Node *l = 0, *r = 0;
  int lo, hi, mset = inf, madd = 0, val = -inf;
  Node(int lo,int hi) : lo(lo), hi(hi) {} // Large interval of
      -inf
  Node(vector <int>& v, int lo, int hi) : lo(lo), hi(hi) {
    if (lo + 1 < hi) {
      int mid = lo + (hi - lo)/2;
      l = new Node(v, lo, mid); r = new Node(v, mid, hi);
      val = max(l->val, r->val);
    }
```

```
    else val = v[lo];
  }
  int query(int L, int R) {
    if (R <= lo || hi <= L) return -inf;
    if (L <= lo && hi <= R) return val;
    push();
    return max(l->query(L, R), r->query(L, R));
  }
  void set(int L, int R, int x) {
    if (R <= lo || hi <= L) return;
    if (L <= lo && hi <= R) mset = val = x, madd = 0;
    else {
      push(), l->set(L, R, x), r->set(L, R, x);
      val = max(l->val, r->val);
    }
  }
  void add(int L, int R, int x) {
    if (R <= lo || hi <= L) return;
    if (L <= lo && hi <= R) {
      if (mset != inf) mset += x;
      else madd += x;
      val += x;
    }
    else {
      push(), l->add(L, R, x), r->add(L, R, x);
      val = max(l->val, r->val);
    }
  }
  void push() {
    if (!l) {
      int mid = lo + (hi - lo)/2;
      l = new Node(lo, mid); r = new Node(mid, hi);
    }
    if (mset != inf)
      l->set(lo, hi, mset), r->set(lo, hi, mset), mset = inf;
    else if (madd)
      l->add(lo, hi, madd), r->add(lo, hi, madd), madd = 0;
  }
};
```

persistentDSU.hh

Description: Disjoint-set data structure with undo. If undo is not needed, skip st, time() and rollback().

Usage: `int t = uf.time(); ...; uf.rollback(t);`

Time: $\mathcal{O}(\log(N))$

21 lines

```
struct DSU {
  vector <int> e; vector <pair <int, int>> st;
  DSU (int n) : e(n, -1) {}
  int size(int x) { return -e[find(x)]; }
  int find(int x) { return e[x] < 0 ? x : find(e[x]); }
  int time() { return sz(st); }
  void rollback(int t) {
    for (int i = time(); i --> t;)
      e[st[i].first] = st[i].second;
    st.resize(t);
  }
  bool join(int a, int b) {
    a = find(a), b = find(b);
    if (a == b) return false;
    if (e[a] > e[b]) swap(a, b);
    st.push_back({a, e[a]});
    st.push_back({b, e[b]});
    e[a] += e[b]; e[b] = a;
    return true;
  }
};
```

intervalContainer.hh

Description: Add and remove intervals from a set of disjoint intervals. Will merge the added interval with any overlapping intervals in the set when adding. Intervals are [inclusive, exclusive).

Time: $\mathcal{O}(\log N)$

25 lines

```
using pii = pair <int, int>;

set <pii>::iterator addInterval(set <pii>& is, int L, int R) {
  if (L == R) return is.end();
  auto it = is.lower_bound({L, R}), before = it;
  while (it != is.end() && it->first <= R) {
    R = max(R, it->second);
    before = it = is.erase(it);
  }
  if (it != is.begin() && (--it)->second >= L) {
    L = min(L, it->first);
    R = max(R, it->second);
    is.erase(it);
  }
  return is.insert(before, {L,R});
}

void removeInterval(set <pii>& is, int L, int R) {
  if (L == R) return;
  auto it = addInterval(is, L, R);
  auto r2 = it->second;
  if (it->first == L) is.erase(it);
  else (int&)it->second = L;
  if (R != r2) is.emplace(R, r2);
}
```

Mathematics (2)

2.1 Equations

$$ax^2 + bx + c = 0 \Rightarrow x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The extremum is given by $x = -b/2a$.

$$\begin{aligned} ax + by = e \\ cx + dy = f \end{aligned} \Rightarrow \begin{aligned} x &= \frac{ed - bf}{ad - bc} \\ y &= \frac{af - ec}{ad - bc} \end{aligned}$$

In general, given an equation $Ax = b$, the solution to a variable x_i is given by

$$x_i = \frac{\det A_i'}{\det A}$$

where A_i' is A with the i'th column replaced by b.

2.2 Recurrences

If $a_n = c_1 a_{n-1} + \cdots + c_k a_{n-k}$, and $r_1, \ldots, r_k$ are distinct roots of $x^k - c_1 x^{k-1} - \cdots - c_k$, there are $d_1, \ldots, d_k$ s.t.

$$a_n = d_1 r_1^n + \cdots + d_k r_k^n.$$

Non-distinct roots r become polynomial factors, e.g. $a_n = (d_1 n + d_2) r^n$.

2.3 Trigonometry

$$\sin(v + w) = \sin v \cos w + \cos v \sin w$$
$$\cos(v + w) = \cos v \cos w - \sin v \sin w$$

$$\tan(v + w) = \frac{\tan v + \tan w}{1 - \tan v \tan w}$$
$$\sin v + \sin w = 2 \sin \frac{v + w}{2} \cos \frac{v - w}{2}$$
$$\cos v + \cos w = 2 \cos \frac{v + w}{2} \cos \frac{v - w}{2}$$

$$(V + W) \tan(v - w)/2 = (V - W) \tan(v + w)/2$$

where V, W are lengths of sides opposite angles v, w.

$$a \cos x + b \sin x = r \cos(x - \phi)$$
$$a \sin x + b \cos x = r \sin(x + \phi)$$

where $r = \sqrt{a^2 + b^2}$, $\phi = \operatorname{atan2}(b, a)$.

2.4 Geometry

2.4.1 Triangles

Side lengths: a, b, c

Semiperimeter: $p = \dfrac{a + b + c}{2}$

Area: $A = \sqrt{p(p - a)(p - b)(p - c)}$

Circumradius: $R = \dfrac{abc}{4A}$

Inradius: $r = \dfrac{A}{p}$

Length of median (divides triangle into two equal-area triangles): $m_a = \frac{1}{2}\sqrt{2b^2 + 2c^2 - a^2}$

Length of bisector (divides angles in two):

$$s_a = \sqrt{bc \left[1 - \left(\frac{a}{b + c}\right)^2\right]}$$

Law of sines: $\dfrac{\sin \alpha}{a} = \dfrac{\sin \beta}{b} = \dfrac{\sin \gamma}{c} = \dfrac{1}{2R}$

Law of cosines: $a^2 = b^2 + c^2 - 2bc \cos \alpha$

Law of tangents: $\dfrac{a + b}{a - b} = \dfrac{\tan \dfrac{\alpha + \beta}{2}}{\tan \dfrac{\alpha - \beta}{2}}$

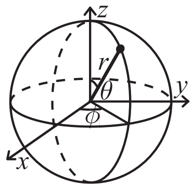
2.4.2 Quadrilaterals

With side lengths a, b, c, d, diagonals e, f, diagonals angle θ, area A and magic flux $F = b^2 + d^2 - a^2 - c^2$:

$$4A = 2ef \cdot \sin \theta = F \tan \theta = \sqrt{4e^2 f^2 - F^2}$$

For cyclic quadrilaterals the sum of opposite angles is $180°$, $ef = ac + bd$, and $A = \sqrt{(p - a)(p - b)(p - c)(p - d)}$.

2.4.3 Spherical coordinates



$$x = r\sin\theta\cos\phi \qquad r = \sqrt{x^2+y^2+z^2}$$
$$y = r\sin\theta\sin\phi \qquad \theta = \operatorname{acos}(z/\sqrt{x^2+y^2+z^2})$$
$$z = r\cos\theta \qquad \phi = \operatorname{atan2}(y,x)$$

2.5 Derivatives/Integrals

$$\frac{d}{dx}\arcsin x = \frac{1}{\sqrt{1-x^2}} \qquad \frac{d}{dx}\arccos x = -\frac{1}{\sqrt{1-x^2}}$$
$$\frac{d}{dx}\tan x = 1+\tan^2 x \qquad \frac{d}{dx}\arctan x = \frac{1}{1+x^2}$$
$$\int \tan ax = -\frac{\ln|\cos ax|}{a} \qquad \int x\sin ax = \frac{\sin ax - ax\cos ax}{a^2}$$
$$\int e^{-x^2} = \frac{\sqrt{\pi}}{2}\operatorname{erf}(x) \qquad \int xe^{ax}dx = \frac{e^{ax}}{a^2}(ax-1)$$

Integration by parts:

$$\int_a^b f(x)g(x)dx = [F(x)g(x)]_a^b - \int_a^b F(x)g'(x)dx$$

2.6 Sums

$$c^a + c^{a+1} + \cdots + c^b = \frac{c^{b+1}-c^a}{c-1}, c \neq 1$$

$$1+2+3+\cdots+n = \frac{n(n+1)}{2}$$
$$1^2+2^2+3^2+\cdots+n^2 = \frac{n(2n+1)(n+1)}{6}$$
$$1^3+2^3+3^3+\cdots+n^3 = \frac{n^2(n+1)^2}{4}$$
$$1^4+2^4+3^4+\cdots+n^4 = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30}$$

2.7 Series

$$e^x = 1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\ldots, (-\infty < x < \infty)$$
$$\ln(1+x) = x-\frac{x^2}{2}+\frac{x^3}{3}-\frac{x^4}{4}+\ldots, (-1 < x \leq 1)$$
$$\sqrt{1+x} = 1+\frac{x}{2}-\frac{x^2}{8}+\frac{2x^3}{32}-\frac{5x^4}{128}+\ldots, (-1 \leq x \leq 1)$$
$$\sin x = x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\ldots, (-\infty < x < \infty)$$
$$\cos x = 1-\frac{x^2}{2!}+\frac{x^4}{4!}-\frac{x^6}{6!}+\ldots, (-\infty < x < \infty)$$

2.8 Probability theory

Let X be a discrete random variable with probability $p_X(x)$ of assuming the value x. It will then have an expected value (mean) $\mu = \mathbb{E}(X) = \sum_x xp_X(x)$ and variance $\sigma^2 = V(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2 = \sum_x (x-\mathbb{E}(X))^2 p_X(x)$ where σ is the standard deviation. If X is instead continuous it will have a probability density function $f_X(x)$ and the sums above will instead be integrals with $p_X(x)$ replaced by $f_X(x)$.

Expectation is linear:

$$\mathbb{E}(aX+bY) = a\mathbb{E}(X) + b\mathbb{E}(Y)$$

For independent X and Y,

$$V(aX+bY) = a^2V(X) + b^2V(Y).$$

2.9 Generating functions

Example of finding generating functions:

$a_0 = 1$

$a_1 = 2$

$a_n = 6a_{n-1} - 8a_{n-2} + n, n \geq 2$

$$G(z) = a_0 + a_1z + \sum_{n=2}^{\infty}(6a_{n-1} - 8a_{n-2} + n)z^n$$

$$G(z) = a_0 + a_1z + 6\sum_{n=2}^{\infty}a_{n-1}z^n - 8\sum_{n=2}^{\infty}a_{n-2}z^n + \sum_{n=2}^{\infty}nz^n$$

$$G(z) = a_0 + a_1z + 6z\sum_{n=1}^{\infty}a_nz^n - 8z^2\sum_{n=0}^{\infty}a_nz^n + \sum_{n=2}^{\infty}nz^n$$

$$G(z) = a_0 + a_1z + 6z(G(z) - a_0) - 8z^2G(z) + \sum_{n=2}^{\infty}nz^n$$

$$G(z) = 1 - 4z + 6zG(z) - 8z^2G(z) + \sum_{n=2}^{\infty}nz^n$$

To calculate $\sum_{n=2}^{\infty}nz^n$ use generating function $B(z)$ for $b_n = (1,1,1,1,...)$:

$$zB'(z) = z(\sum_{n=0}^{\infty}b_nz^n)' = \sum_{n=0}^{\infty}b_nz^n$$

Then we can do:

$$\sum_{n=2}^{\infty}nz^2 = z\sum_{n=2}^{\infty}nz^{n-1} = z(\sum_{n=2}^{\infty}z^n)'$$

$$\sum_{n=2}^{\infty}z^n = \sum_{n=0}^{\infty}z^n - 1 - z = \frac{1}{1-z} - 1 - z = \frac{z^2}{1-z}$$

$$z(\tfrac{z^2}{1-z})' = \tfrac{z^2(2-z)}{(1-z)^2}$$

Then:

$$G(z) = 1 - 4z + 6zG(z) - 8z^22G(z) + \tfrac{z^2(2-z)}{(1-z)^2}$$

$$G(z) = \tfrac{1-6z+11z^2-5z^3}{(1-6z+8z^2)(1-z)^2}$$

Последовательность	Производящая функция в виде ряда	Производящая функция в замкнутом виде
$(1,0,0,\ldots)$	1	1
$(0,0,\ldots,0,1,0,0\ldots)$ (m нулей в начале)	z^m	z^m
$(1,1,1,\ldots)$	$\sum z^n$	$\frac{1}{1-z}$
$(1,0,0,\ldots,0,1,0,0,\ldots0,1,0,0\ldots)$ (повторяется через m)	$\sum z^{nm}$	$\frac{1}{1-z^m}$
$(1,-1,1,-1,\ldots)$	$\sum(-1)^nz^n$	$\frac{1}{1+z}$
$(1,2,3,4,\ldots)$	$\sum(n+1)z^n$	$\frac{1}{(1-z)^2}$
$(1,2,4,8,16,\ldots)$	$\sum 2^nz^n$	$\frac{1}{(1-2z)}$
$(1,r,r^2,r^3,\ldots)$	$\sum r^nz^n$	$\frac{1}{(1-rz)}$
$(\binom{m}{0},\binom{m}{1},\binom{m}{2},\binom{m}{3},\ldots)$	$\sum\binom{m}{n}z^n$	$(1+z)^m$
$(1,\binom{m}{m},\binom{m+1}{m},\binom{m+2}{m},\ldots)$	$\sum\binom{m+n-1}{n}z^n$	$\frac{1}{(1-z)^m}$
$(1,\binom{m+1}{m},\binom{m+2}{m},\binom{m+3}{m},\ldots)$	$\sum\binom{m+n}{n}z^n$	$\frac{1}{(1-z)^{m+1}}$
$(0,1,-\frac{1}{2},\frac{1}{3},-\frac{1}{4},\ldots)$	$\sum\frac{(-1)^{n+1}}{n}z^n$	$\ln(1+z)$
$(1,1,\frac{1}{2},\frac{1}{6},\frac{1}{24},\ldots)$	$\sum\frac{1}{n!}z^n$	e^z
$(1,-\frac{1}{2!}m^2,\frac{1}{4!}m^4,-\frac{1}{6!}m^6,\frac{1}{8!}m^8,\ldots)$	$\sum\frac{1}{(2n)!}m^{(2n)}$	$\cos m$
$(m,-\frac{1}{3!}m^3,\frac{1}{5!}m^5,-\frac{1}{7!}m^7,\frac{1}{9!}m^9,\ldots)$	$\sum\frac{1}{(2n-1)!}m^{(2n-1)}$	$\sin m$