

JakeMate14

Contents

1	Estructuras de Datos	1
1.1	Segment tree	1
2	Graphs	2
3	Math	2
4	Geometry	2
5	Strings	2
6	Flow	2
7	Other	2

1 Estructuras de Datos

1.1 Segment tree

```
1 struct segtree {
2     int size;
3     vector<ll> vv;
4
5     void build(vector<int> &nums) {
6         size = 1;
7         while (size < nums.size()) size *= 2;
8         vv.assign(2 * size, 0);
9         build(nums, 0, 0, size);
10    }
11
12    void build(vector<int> &nums, int x, int lx, int rx) {
13        if (rx - lx == 1) {
14            if (lx < nums.size()) {
15                vv[x] = nums[lx];
16            }
17        } else {
18            int m = (lx + rx) / 2;
19            build(nums, 2 * x + 1, lx, m);
20            build(nums, 2 * x + 2, m, rx);
21            vv[x] = vv[2 * x + 1] + vv[2 * x + 2];
22        }
23    }
24
25    void set(int i, int v) {
26        set(i, v, 0, 0, size);
27    }
28
29    void set(int i, int v, int x, int lx, int rx) {
30        if (rx - lx == 1) {
31            vv[x] = v;
32        } else {
33            int m = (lx + rx) / 2;
34            if (i < m) {
35                set(i, v, 2*x+1, lx, m);
36            } else {
37                set(i, v, 2*x+2, m, rx);
38            }
39            vv[x] = vv[2*x+1] + vv[2*x+2];
40        }
41    }
42 }
```

```
40     }
41 }
42
43 ll sum(int l, int r) {
44     return sum(l, r, 0, 0, size);
45 }
46
47 ll sum(int l, int r, int x, int lx, int rx) {
48     if (r <= lx) return 0;
49     if (l >= rx) return 0;
50     if (lx >= l && rx <= r) return vv[x];
51     int m = (lx + rx) / 2;
52     ll s1 = sum(l, r, 2*x+1, lx, m);
53     ll s2 = sum(l, r, 2*x+2, m, rx);
54     return s1 + s2;
55 }
56 };
```

2 Graphs

3 Math

4 Geometry

5 Strings

6 Flow

7 Other