

Сканирование



cut - кон-бо отрезка;

struct event {

int time, type;

int id;

}

События

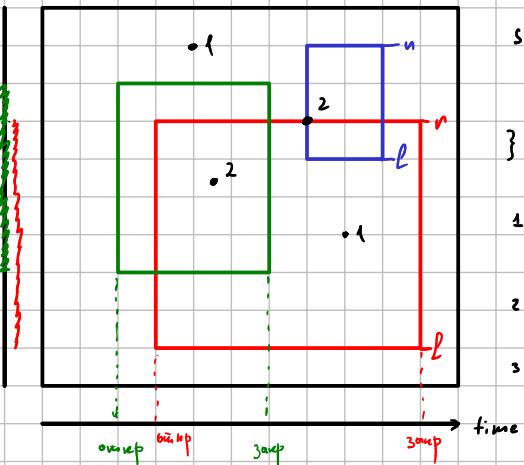
1 - отр. начало

-1 - отр. конец

0 -毛ка

конец
запроса
б вход.

Покрытие в промежутке



struct {

int time, type, id;

int l, r;

**}
y для毛ка**

1. Отр.

st.upd(e.l, e.r, +1)

2. Закр.

st.upd(e.l, e.r, -1)

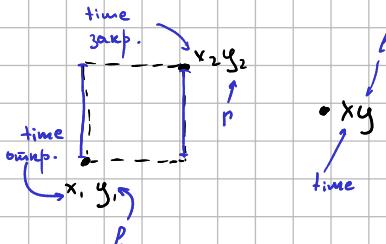
3. Марка

ans[e.id] = st.get(e.l);

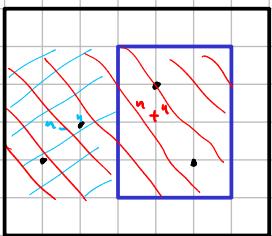
Количество точек в отрезке

```
C++
```

```
int n, m; cin >> n >> m;
vector<event> line;
vector<int> ans(m, 0);
while (n--) {
    int x; cin >> x;
    line.push_back({x, 0, 0});
}
for (int i = 0; i < m; i++) {
    int l, r; cin >> l >> r;
    line.push_back({l, -1, i});
    line.push_back({r, 1, i});
}
sort(line.begin(), line.end());
[<] (event lhs, event rhs) {
    return make_pair(lhs.time, lhs.type) <
        make_pair(rhs.time, rhs.type);
};
int cnt = 0;
for (event e : line) {
    cnt += !e.type;
    ans[e.id] += cnt * e.type;
}
for (int x : ans)
    cout << x << ' ';
```



Кон-бо毛ка в промежутке.



st[i] - кон-бо в. на i-й毛ка

в момент \leq сейчас

1. Марка

st.upd(e.l, e.r)

[; ; ;]

-1 0 1

2. Отр.

ans[e.id] = -st.get(e.l, e.r)

[; ; ;]

3. Закр.

ans[e.id] += st.get(e.l, e.r)

[; ; ;]

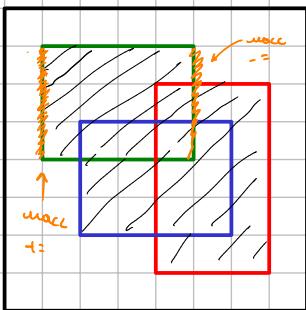
ans[e.id] += e.type * st.get(e.l, e.r)

Сжатие координат

```
C++
```

```
vector<int> sv = vec;
sort(sv.begin(), sv.end());
unordered_map<int, int> mp;
for (int &x: sv) {
    if (!mp[x])
        mp[x] = mp.size();
}
for (int &x : vec)
    x = mp[x];
```

Одномерные промежутки.



1. ans += get(w)

2. upd

Do на

1. ищем.

→ кон-бо не 0

= бит - [кон-бо 0]

2. кон-бо ищем

3. масс +=

Решение

