SOLVE LATER

Happy segments

Attempted by: **3224** / Accuracy: **84%** / Maximum Score: **30** / 96 Votes

Tag(s): Advanced Data Structures, Data Structures, Lazy Propagation in Interval/Segment Trees

Lets fix a number x and all its occurrences. Suppose that the number x is one of the numbers that make some particular segment $[l, r]$ be bad. Then one of the following situations should happen: either there are from 1 to $x - 1$ or more than $x + 1$ occurrences in this segment. That means that for the fixed l there are two segments of banned values for r. Actually, for any possible l lying between two consectuvie x's, those two banned segments for r will be the same. It can be shown as banned rectangles on a plane that point, for fixed $l1, r1, l2, r2$, l, r is bad segment if $l1 \le l \le l2, r1 \le r \le r2$ (those rectangle).

Lets answer to queries offline. Construct all such rectangles for all numbers by scan-line, just updatin ranges. And for fixed l check value on rth positions.

Author Solution by Narkhan Kamzabek

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

#define mp make_pair
#define pb push_back
#define f first
#define s second
#define ll long long
#define forn(i, a, b) for(int i = (a); i <= (b); ++i)
#define forev(i, b, a) for(int i = (b); i >= (a); --i)
#define VAR(v, i) __typeof( i) v=(i)
#define forit(i, c) for(VAR(i, (c).begin()); i != (c).end(); ++i)
#define all(x) (x).begin(), (x).end()
#define sz(x) ((int)(x).size())
#define file(s) freopen(s".in","r",stdin); freopen(s".out","w",stdout);

using namespace std;

const int maxn = (int)1e6 + 100;
const int mod = (int)1e9 + 7;

#define inf mod

typedef long double ld;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef vector<int> vi;
typedef vector<ll> Vll;
typedef vector<pair<int, int> > vpii;
typedef vector<pair<ll, ll> > vpll;

int n, m;

vi g[maxn];

vpii add[maxn], del[maxn];

inline void Add(int l1, int r1, int l2, int r2) {
    assert(1 <= l1 && l1 <= r1 && r1 <= l2 && l2 <= r2 && r2 <= n);
    add[l1].emplace_back(l2, r2);
```

```
    del[r1].emplace_back(l2, r2);
}

inline void process(int c, vi pos){
        for(auto i = 0; i + 2 < sz(pos); i++){
                int l1 = pos[i] + 1, r1 = pos[i + 1], l2 = pos[i + 1], r2
= ((i + c < pos.size()) ? pos[i + c] - 1 : n);
         if(c != 1) Add(l1, r1, l2, r2);
                if(i + c + 2 < sz(pos)){
                        int l3 = pos[i + c + 1], r3 = n;
                        Add(l1, r1, l3, r3);
                }
        }
}

int q, t[maxn];
bool ans[maxn];
vpii qu[maxn];

inline void upd(int pos, int val){
        for(; pos < maxn; pos |= (pos + 1))
                t[pos] += val;
}
inline int get(int r){
    int res = 0;
    for(; r >= 0; r = (r & (r + 1)) - 1) res += t[r];
    return res;
}
inline void upd(int l, int r, int x){
    upd(l, x);
    upd(r + 1, -x);
}

int main () {

        scanf("%d%d", &n, &m);

        forn(i, 1, n){
                int a;
```

```
                scanf("%d", &a);
                assert(1 <= a && a <= m);
                g[a].pb(i);
        }

        forn(i, 1, m){
                int h;
                scanf("%d", &h);
                if(g[i].empty()) continue;
                g[i].insert(g[i].begin(), 0);
                g[i].insert(g[i].end(), n + 1);
                process(h, g[i]);
        }

        scanf("%d", &q);

        forn(i, 1, q){
                int l, r;
                scanf("%d%d", &l, &r);
                qu[l].pb(mp(r, i));
        }

        forn(i, 1, n){
                for(auto seg : add[i]) upd(seg.f, seg.s, 1);
                for(auto it : qu[i]) ans[it.s] = (get(it.f) ? 0 : 1);
                for(auto seg : del[i]) upd(seg.f, seg.s, -1);
        }

        forn(i, 1, q) printf("%d\n", ans[i]);
}
```

Tester Solution by Amirreza Poorakhavan

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

```

```cpp
#define mp make_pair
#define pb push_back
#define f first
#define s second
#define ll long long
#define forn(i, a, b) for(int i = (a); i <= (b); ++i)
#define forev(i, b, a) for(int i = (b); i >= (a); --i)
#define VAR(v, i) __typeof( i) v=(i)
#define forit(i, c) for(VAR(i, (c).begin()); i != (c).end(); ++i)
#define all(x) (x).begin(), (x).end()
#define sz(x) ((int)(x).size())
#define file(s) freopen(s".in","r",stdin); freopen(s".out","w",stdout);

using namespace std;

const int maxn = (int)1e6 + 100;
const int mod = (int)1e9 + 7;

#define inf mod

typedef long double ld;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef vector<int> vi;
typedef vector<ll> Vll;
typedef vector<pair<int, int> > vpii;
typedef vector<pair<ll, ll> > vpll;

int n, m;

vi g[maxn];

vpii add[maxn], del[maxn];

inline void Add(int l1, int r1, int l2, int r2) {
    assert(1 <= l1 && l1 <= r1 && r1 <= l2 && l2 <= r2 && r2 <= n);
    add[l1].emplace_back(l2, r2);
    del[r1].emplace_back(l2, r2);
}
```

```
42.
43. inline void process(int c, vi pos){
44.         for(auto i = 0; i + 2 < sz(pos); i++){
45.                 int l1 = pos[i] + 1, r1 = pos[i + 1], l2 = pos[i + 1], r2
    = ((i + c < pos.size()) ? pos[i + c] - 1 : n);
46.          if(c != 1) Add(l1, r1, l2, r2);
47.                 if(i + c + 2 < sz(pos)){
48.                         int l3 = pos[i + c + 1], r3 = n;
49.                         Add(l1, r1, l3, r3);
50.                 }
51.         }
52. }
53.
54. int q, t[maxn];
55. bool ans[maxn];
56. vpii qu[maxn];
57.
58. inline void upd(int pos, int val){
59.         for(; pos < maxn; pos |= (pos + 1))
60.                 t[pos] += val;
61. }
62. inline int get(int r){
63.     int res = 0;
64.     for(; r >= 0; r = (r & (r + 1)) - 1) res += t[r];
65.     return res;
66. }
67. inline void upd(int l, int r, int x){
68.     upd(l, x);
69.     upd(r + 1, -x);
70. }
71.
72. int main () {
73.
74.         scanf("%d%d", &n, &m);
75.
76.         forn(i, 1, n){
77.                 int a;
78.                 scanf("%d", &a);
79.                 assert(1 <= a && a <= m);
```

```
                g[a].pb(i);
        }

        forn(i, 1, m){
                int h;
                scanf("%d", &h);
                if(g[i].empty()) continue;
                g[i].insert(g[i].begin(), 0);
                g[i].insert(g[i].end(), n + 1);
                process(h, g[i]);
        }

        scanf("%d", &q);

        forn(i, 1, q){
                int l, r;
                scanf("%d%d", &l, &r);
                qu[l].pb(mp(r, i));
        }

        forn(i, 1, n){
                for(auto seg : add[i]) upd(seg.f, seg.s, 1);
                for(auto it : qu[i]) ans[it.s] = (get(it.f) ? 0 : 1);
                for(auto seg : del[i]) upd(seg.f, seg.s, -1);
        }

        forn(i, 1, q) printf("%d\n", ans[i]);
}
```


COMMENTS (8)

SORT BY: Relevance

Join Discussion...

Cancel Post

Arnaud Desombre Edited 2 days ago

This editorial is written in markdown, which Hackerearth does not support....
Tip: using an online markdown editor (for example https://stackedit.io/app#), copy the text of the editorial to get the desired output. (It would still be a poor editorial though.)

8 votes ● Reply ● Message ● Permalink

Shaurya Manhar 2 days ago

thanks..

0 votes ● Reply ● Message ● Permalink

Pritam Oberoy a day ago

Worst editorial ever

8 votes ● Reply ● Message ● Permalink

Eklavya a day ago

I solve hackerearth problems to learn from editorials for the problems that i can't solve. but these kind of editorials disappoints me . please anyone help me to understand the solution

7 votes ● Reply ● Message ● Permalink

Divyansh Garg a day ago

can anyone provide a better approch or just explain how to solve this problem?

3 votes ● Reply ● Message ● Permalink

bpc Edited a day ago

Can an example be provided for this editorial and also what is time complexity for the above problem?

3 votes ● Reply ● Message ● Permalink

Arnaud Desombre Edited 10 hours ago

Is this a joke?
1/ Author Solution by Narkhan Kamzabek and Tester Solution by Amirreza Poorakhavan are EXACTLY the same (they probably failed to read the plagiarism rule).
2/ The editorial is not well formatted (Hackerearth does not support markdown).
3/ The editorial is WRONG:

"Then one of the following situations should happen: either there are from 1 to $x - 1$ or more than $x + 1$ occurrences in this segment" should be "Then one of the following situations should happen: either there are from 1 to $h(x) - 1$ or more than $h(x) + 1$ occurrences in this segment".
4/ I'm not sure anyone can understand this editorial!
This problem was interesting & fun and deserves a well designed editorial. I'd be grateful for anyone who could provide a link to such editorial. Thank you!

▲ 1 vote ● Reply ● Message ● Permalink

Dheeraj Poonia 17 hours ago

worst editorial make complete contest SHIT

▲ 0 votes ● Reply ● Message ● Permalink

+1-650-461-4192

contact@hackerearth.com