CHALLENGES PRACTICE COMPANIES

All Tracks > Data Structures > Advanced Data Structures > Segment Trees > Problem



Inverse Subarray

Attempted by: 1163 / Accuracy: 53% / Maximum Score: 30 /

Tag(s): Easy-Medium, Segment Trees, Two-pointer, approved

PROBLEM EDITORIAL

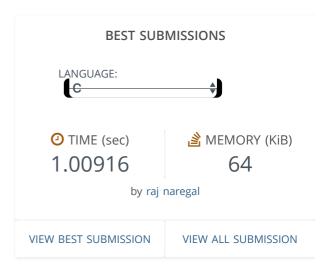
MY SUBMISSIONS

ANALYTICS

The problem "Inverse Subarray" doesn't have any editorial. You can contribute it by sending editorial in markdown format to moderator@hackerearth.com.

Author Solution by Shubham Garg

- 1. /*masterwayne*/
- 2. #include <bits/stdc++.h>
- 3. using namespace std;
- 4. #define sc(x) scanf("%d",&x)
- 5. #define sc2(x,y) scanf("%d %d",&x,&y)
- 6. #define sc3(x,y,z) scanf("%d %d
 %d",&x,&y,&z)
- 7. #define pf(x) printf("%d",x)
- 8. #define pf2(x,y) printf("%d %d",x,y)
- 9. #define pf3(x,y,z) printf("%d %d
 %d",x,y,z)
- 10. #define fr(i,x,n) for(int
 i=x;i<n;i++)</pre>
- 11. #define fre(i,x,n) for(int
 i=x;i<=n;i++)</pre>
- 12. #define fb(i,x,n) for(int i=n1;i>=x;i--)
- 13. #define fbe(i,x,n) for(int
 i=n;i>=x;i--)
- 14. #define pfn() printf("\n")
- 15. #define pfs() printf(" ")
- 16. #define pb push_back
- 17. #define f_in(st)



CONTRIBUTOR





THIS PROBLEM WAS ASKED IN

Alacriti Fresher Hiring Challenge

SOCIAL SHARE

f y in G+

```
freopen(st, "r", stdin)
18. #define f_out(st)
   freopen(st, "w", stdout)
19.
20. int tree[1000001];
21. int final[1000001];
22. void build(int i,int start,int end)
23. {
24.
            if(start==end)
25.
            {
26.
                     tree[i]=final[start];
27.
                     return ;
28.
            }
29.
            else
30.
31.
                     int mid =
   (start+end)/2;
32.
   build(2*i+1,start,mid);
33.
   build(2*i+2, mid+1, end);
34.
                     tree[i] =
   max(tree[2*i+1],tree[2*i+2]);
35.
            }
36. }
37. int query(int node,int 1,int r,int
   lr,int rr)
38. {
39.
            if(rr<1 || lr > r)
40.
                     return INT MIN;
41.
            else
42.
            {
43.
                     if(lr<=l&&r<=rr)
44.
                              return
   tree[node];
45.
   max(query(2*node+1,1,
   (1+r)/2, lr, rr), query(2*node+2,
   ((1+r)/2)+1,r,lr,rr));
46.
            }
47. }
48. void solve(string input, string
   output)
49. {
50.
            f_in(input.c_str());
51.
            f_out(output.c_str());
52.
            int t;
53.
            cin>>t;
54.
            while(t--)
55.
```

```
56.
                     int n;
57.
                     cin>>n;
                     vector< pair<int,int>
58.
   > vec;
59.
   memset(tree, 0, sizeof(tree));
60.
                     for(int i=0;i<n;i++)</pre>
61.
62.
                              int x;
63.
                              cin>>x;
64.
   vec.push_back({x,i});
65.
                     }
66.
   sort(vec.begin(),vec.end());
67.
                     for(int i=0;i<n;i++)</pre>
68.
   final[i]=vec[i].second;
69.
                     build(0,0,n-1);
70.
                     int ans = 1;
71.
                      for(int i=n-1;i>=0;i-
   -)
72.
                      {
73.
                               int first =
   vec[i].second;
74.
                              int second =
   query(0,0,n-1,0,i-1);
75.
   if(second>first)
76.
                                       ans =
   max(ans, second-first+1);
77.
78.
                     cout << ans << endl;
79.
            }
80. }
81. int main()
82. {
83.
            solve("in00.txt","out00.txt")
            solve("in01.txt","out01.txt")
84.
85.
            solve("in02.txt","out02.txt")
86.
            solve("in03.txt","out03.txt")
87.
            solve("in04.txt","out04.txt")
88.
            solve("in05.txt","out05.txt")
89.
            solve("in06.txt","out06.txt")
```

