SOLVE LATER

Inverse Subarray

Attempted by: **1163** / Accuracy: **53%** / Maximum Score: **30** /
2 Votes

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

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++)
11. #define fre(i,x,n) for(int
    i=x;i<=n;i++)
12. #define fb(i,x,n) for(int i=n-
    1;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)
```

BEST SUBMISSIONS

LANGUAGE: C

TIME (sec)	MEMORY (KiB)
1.00916	64

by raj naregal

VIEW BEST SUBMISSION VIEW ALL SUBMISSION

CONTRIBUTOR

AUTHOR
Shubham Garg

TESTER
Anand Jaisingh

THIS PROBLEM WAS ASKED IN

Alacriti CHALLENGE NAME
Alacriti Fresher Hiring Challenge

SOCIAL SHARE

```
freopen(st,"r",stdin)
#define f_out(st)
freopen(st,"w",stdout)

int tree[1000001];
int final[1000001];
void build(int i,int start,int end)
{
        if(start==end)
        {
                tree[i]=final[start];
                return ;
        }
        else
        {
                int mid =
(start+end)/2;

build(2*i+1,start,mid);

build(2*i+2,mid+1,end);
                tree[i] =
max(tree[2*i+1],tree[2*i+2]);
        }
}
int query(int node,int l,int r,int
lr,int rr)
{
        if(rr<l || lr > r)
                return INT_MIN;
        else
        {
                if(lr<=l&&r<=rr)
                        return
tree[node];
                return
max(query(2*node+1,l,
(l+r)/2,lr,rr),query(2*node+2,
((l+r)/2)+1,r,lr,rr));
        }
}
void solve(string input,string
output)
{
        f_in(input.c_str());
        f_out(output.c_str());
        int t;
        cin>>t;
        while(t--)
        {
```

```
                int n;
                cin>>n;
                vector< pair<int,int> > vec;
memset(tree,0,sizeof(tree));
                for(int i=0;i<n;i++)
                {
                        int x;
                        cin>>x;
vec.push_back({x,i});
                }
sort(vec.begin(),vec.end());
                for(int i=0;i<n;i++)
final[i]=vec[i].second;
                build(0,0,n-1);
                int ans = 1;
                for(int i=n-1;i>=0;i--)
                {
                        int first = vec[i].second;
                        int second = query(0,0,n-1,0,i-1);
if(second>first)
                                ans = max(ans,second-first+1);
                }
                cout<<ans<<endl;
        }
}
int main()
{
        solve("in00.txt","out00.txt");
        solve("in01.txt","out01.txt");
        solve("in02.txt","out02.txt");
        solve("in03.txt","out03.txt");
        solve("in04.txt","out04.txt");
        solve("in05.txt","out05.txt");
        solve("in06.txt","out06.txt")
```

```
    ;
90.         solve("in07.txt","out07.txt")
    ;
91.         solve("in08.txt","out08.txt")
    ;
92.         solve("in09.txt","out09.txt")
    ;
93.         return 0;
94. }
```

COMMENTS (4) SORT BY: Relevance

Join Discussion...

Cancel Post

jaydeep hi 5 months ago

Can anyone explain ?

▲ 2 votes ● Reply ● Message ● Permalink

Sahitya Patel 6 months ago

Easy 2 pointers method with a little bit optimization : https://www.hackerearth.com/submission/9248936/

▲ 1 vote ● Reply ● Message ● Permalink

hardik agrawal 2 months ago

I used a different approach than editorial. Try taking every index of array as the starting point and apply binary search for the ending index of the subarray starting at index i. Let's say you select mid as the ending index than if there is some element in range [mid,n] which is less than arr[i] than you can set lo as mid. since you can guarantee to increase subarray length. else set hi = mid-1.
my code :-
/*input
1
5
1 2 3 4 5
*/
#include <bits/stdc++.h>
#include<stdio.h>