

Equal Stack.

STDIN

Function

5 3 4

h1[] size n1 = 5, h2[] size n2 = 3, h3[] size n3 = 4

3 2 1 1 1

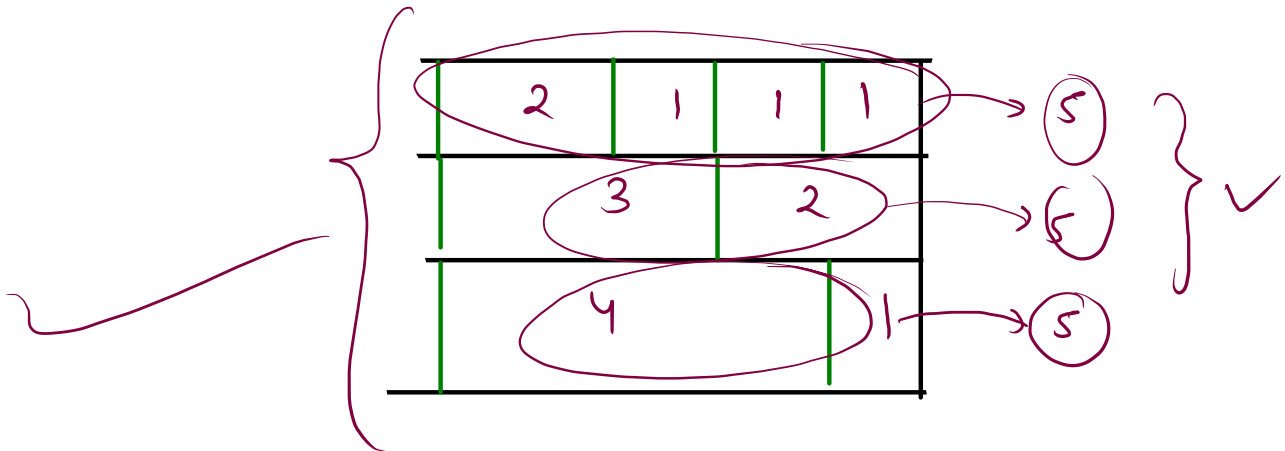
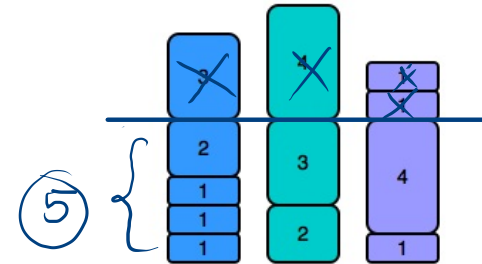
h1 = [3, 2, 1, 1, 1]

4 3 2

h2 = [4, 3, 2]

1 1 4 1

h3 = [1, 1, 4, 1]



~~3~~ 2 1 1 1 } $s_1 \rightarrow$ ~~8~~(5)
~~1~~ 3 2 } $s_2 \rightarrow$ ~~9~~(5)
~~X~~ ~~X~~ 4 1 } $s_3 \rightarrow$ ~~X~~~~6~~(5)

~~X~~ $s_1 > s_2$ and $s_1 > s_3$
~~X~~ $s_2 > s_1$ & $s_2 > s_3$
~~X~~ $s_3 > s_1$ & $s_3 > s_2$

not true

! ($s_1 == s_2$ & $s_2 == s_3$) we have ans.

remove

$s_1 > s_2$ & $s_1 > s_3 \rightarrow s_1 = \text{first ele.}$

$s_2 > s_1$ & $s_2 > s_3 \rightarrow s_2 = \text{first ele.}$

$s_3 > s_1$ & $s_3 > s_2 \rightarrow s_3 = \text{first element}$

~~8~~ 2 1 1 1

~~4~~ 3 2

~~X~~ 1 1 4 1

81 → ~~8~~ 5

82 → ~~9~~ 5

83 → ~~8~~ 7

8 = 5 = 8

8 ≥ 5 ✓

8 ≥ 5 ✓

```
while( !((s1 == s2) && (s1 == s3))){
    if(s1 >= s2 && s1 >= s3){
        s1 -= h1.remove(0);
    }
    else if(s2 >= s1 && s2 >= s3){
        s2 -= h2.remove(0);
    }
    else if(s3 >= s1 && s3 >= s2){
        s3 -= h3.remove(0);
    }
}
```

~~A~~ B C D E ~~F~~

)

(F)

TC = ?

$O(n)$



X

(B)
TC

$O(1)$

```
public static int calcSum(List<Integer> list){
    int sum = 0;
    for(int e : list){
        sum += e;
    }
    return sum;
}

public static int equalStacks(List<Integer> h1, List<Integer> h2, List<Integer> h3) {
    // Write your code here
    int s1 = calcSum(h1);
    int s2 = calcSum(h2);
    int s3 = calcSum(h3);

    while( !((s1 == s2) && (s1 == s3))){
        if(s1 >= s2 && s1 >= s3){
            s1 -= h1.remove(0);
        }
        else if(s2 >= s1 && s2 >= s3){
            s2 -= h2.remove(0);
        }
        else if(s3 >= s1 && s3 >= s2){
            s3 -= h3.remove(0);
        }
    }

    return s1;
}
```

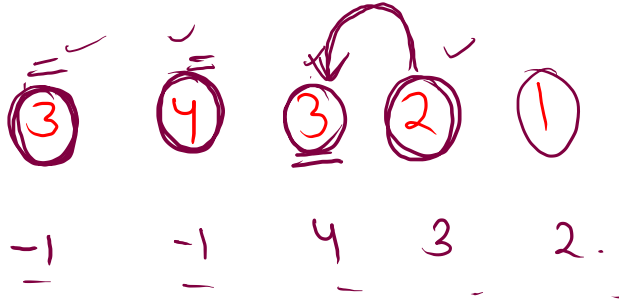
Next greater element on left 1

Sample Input 0

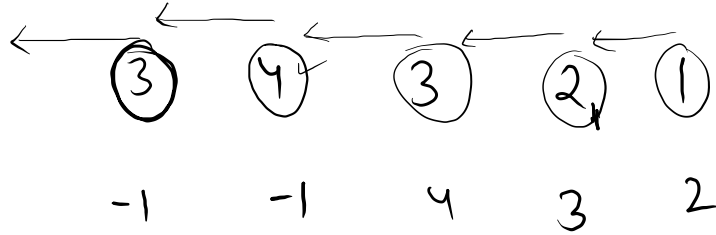
```
5  
3 4 3 2 1
```

Sample Output 0

```
-1 -1 4 3 2
```

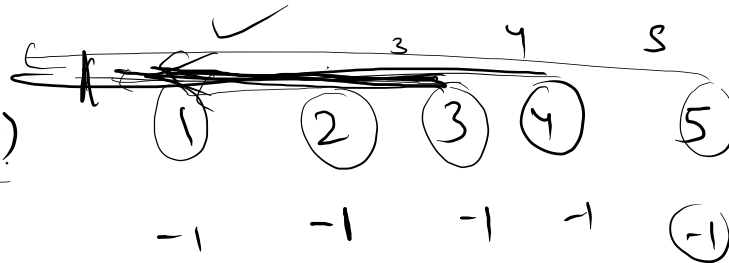


brute force.

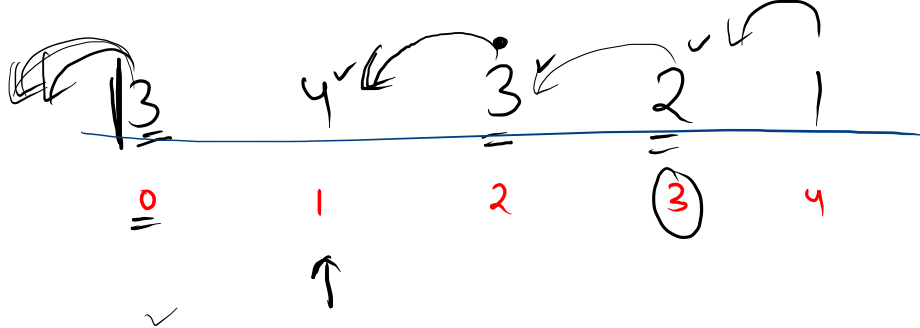


$$1 + 2 + 3 + 4 + 5$$

$$\sum_n = O(n^2)$$



A



-1	-1	4	3	2
----	----	---	---	---

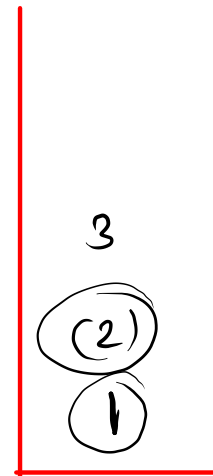
✓

$$A[\text{peek}] > \text{curr.}(A[i])$$

A[peek]

$$4 > 3$$

$$A[i] > 3$$



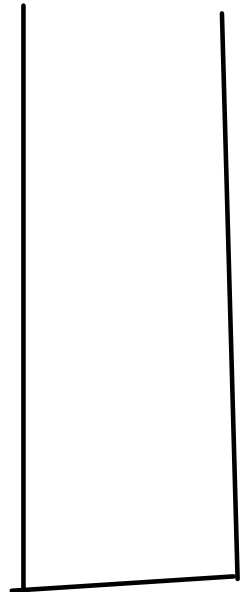
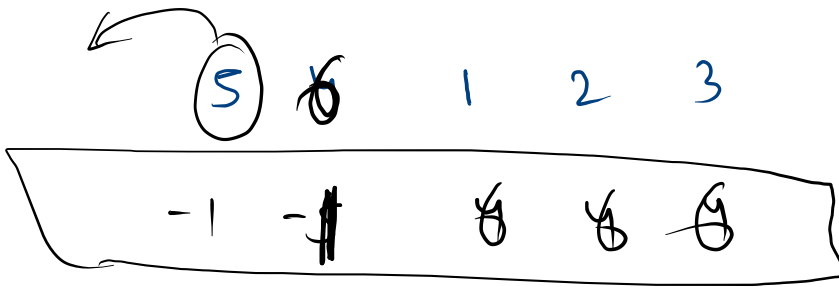
curr
1
(4)

$$A[2] > 2$$

$$3 > 2$$

$$A[3] > 1$$

$$2 > 1$$



3.
-1

4
-1

6
-1

2
6 (3)

1
2

2
5
6

3
6
6

```
int [] ans = new int[A.length];  
ans[0] = -1;  
Stack<Integer> st = new Stack<>();  
st.push(0);    //I am inserting index no
```

```
for(int i = 1; i < ans.length; i++){  
    while(A[st.peek()] < A[i]){  
        st.pop();  
    }  
    if(st.size() == 0){  
        ans[i] = -1;  
    }  
    else {  
        ans[i] = A[st.peek()];  
    }  
    st.push(i);  
}
```

\checkmark
 $A[2] < A[6]$

3
18

2

```

4 public class Solution {
5     public static int [] nge1(int [] A){
6         int [] ans = new int[A.length];
7         ans[0] = -1;
8         Stack<Integer> st = new Stack<>();
9         st.push(0);    //I am inserting index not value
10
11         for(int i = 1; i < ans.length; i++){
12             while(st.size() != 0 && A[st.peek()] <= A[i]){
13                 st.pop();
14             }
15             if(st.size() == 0){
16                 ans[i] = -1;
17             }
18             else {
19                 ans[i] = A[st.peek()];
20             }
21             st.push(i);
22         }
23
24         return ans;
25     }
26
27     public static void main(String[] args) {
28         Scanner scn = new Scanner(System.in);
29         int n = scn.nextInt();
30         int [] A = new int[n];
31         for(int i = 0; i < n; i++){
32             A[i] = scn.nextInt();
33         }
34
35         int [] ans = nge1(A);    //next greater element on left
36         for(int e : ans){
37             System.out.print(e + " ");
38         }
39     }
40 }
41

```

$\underline{5}$ $\textcircled{4}$ $\underline{3}$ 2 6

$O(n^2)$

for ($i=0 \rightarrow n$)

for ($j=0 \rightarrow n$)

5 4 3 2 1 6

in total.

$O(n)$

Online Stock Spanner

previous index

Sample Input 0

7
100 80 60 70 60 75 85



Sample Output 0

1 1 1 2 1 4 6

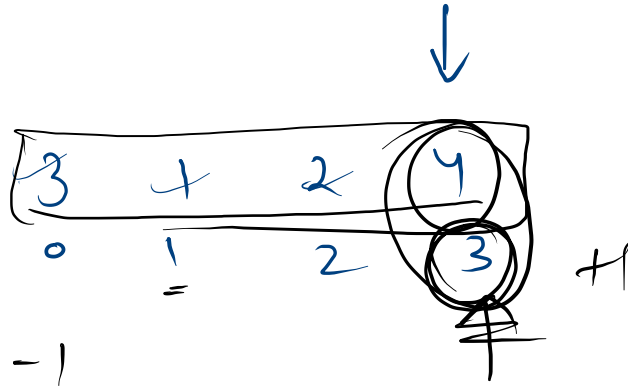
next greater
on left

day = ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~
6

stock prices

← consec.
smaller or equal

$$5 - 1 = 4$$



$$A[2] \leq A[3]$$

$$2 \leq 4$$

$$A[0] \leq A[3]$$

$$3 \leq 4$$

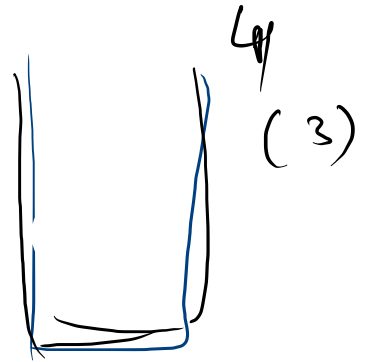
$$A[1] \leq A[2]$$

$$A[\text{prev}] \leq A[i]$$

pop

$$A[0] \leq A[1]$$

$$3 \leq 1$$



$$\text{idx} + 1$$

```

3
4 public class Solution {
5     public static int [] stockSpan(int [] A){
6         int [] ans = new int[A.length];
7         ans[0] = 1;
8         Stack<Integer> st = new Stack<>();
9         st.push(0);    //I am inserting index not value
10
11         for(int i = 1; i < ans.length; i++){
12             while(st.size() != 0 && A[st.peek()] <= A[i]){
13                 st.pop();
14             }
15             if(st.size() == 0){
16                 ans[i] = i + 1;
17             }
18             else {
19                 ans[i] = i-st.peek();
20             }
21
22             st.push(i);
23         }
24
25
26         return ans;
27     }
28     public static void main(String[] args) {
29         Scanner scn = new Scanner(System.in);
30         int n = scn.nextInt();
31         int [] A = new int[n];
32         for(int i = 0; i < n; i++){
33             A[i] = scn.nextInt();
34         }
35
36         int [] ans = stockSpan(A);    //next greater element on left
37         for(int e : ans){
38             System.out.print(e + " ");
39         }
40     }

```