

## Find The Index of Rotation

Given a sorted and rotated array. Find the index at which the array is rotated using binary search.

ans = 1 (idx).

eg.

6 7 1 2 3 4 5  
0 1 2 3 4 5 6

$$id_X = -1 \cdot \beta^2$$

ignore sorted part  
but first store

first store  
potential answer m

۱۰

 $6 < 2$ 

$m \rightarrow j$   
sorted.

1 < 2

\* smallest no. idx in RS array.  
 \* largest " " " " " " " " " "

ans = 3

smallest idx

6	7	9	2	3	4	5
0	1	2	3	4	5	6

j  
i

m

9 < 2

```

4 public class Solution {
5     public static void main(String[] args) {
6         Scanner scn = new Scanner(System.in);
7         int n = scn.nextInt();
8         int [] A = new int[n];
9         for(int i = 0; i < n; i++){
10             A[i] = scn.nextInt();
11         }
12         int i = 0;
13         int j = n-1;
14         int idx = -1; //ans
15         int min = Integer.MAX_VALUE;
16         while(i<=j){
17             int m = (i+j)/2;
18             if(A[i] <= A[m]){ //left sorted
19                 if(A[i] < min){
20                     min = A[i];
21                     idx = i;
22                 }
23                 i = m + 1;
24             }
25             else if(A[m] <= A[j]){ //right sorted
26                 if(A[m] < min){
27                     min = A[m];
28                     idx = m;
29                 }
30                 j = m - 1;
31             }
32         }
33         System.out.println(idx-1);
34     }
35 }

```

6 7 9  
 0 1 2  
 i

2 3 4 5  
 3 4 5 6  
 j

m

min = ~~0~~ 2  
 idx = ~~1~~ 3

# Maximum Product Subarray 2

2      3      -2      4

2      2  
2

3      3  
2 3

-2      -12  
2 3 -2

-48  
2 3 -2 4

3      3  
3 -6  
3 -2

-24  
3 -2 4

-2      -2  
-2      -8  
4  
4      4

$$\begin{aligned}
 \max &= \cancel{2} \times \cancel{144} \times 576 \times 0 \\
 \text{prev max} &= \cancel{2} \times \cancel{144} \times 576 \times \textcircled{-48} \\
 \text{prev min} &= \cancel{2} \times \cancel{144} \times \cancel{576} \times \textcircled{-48}
 \end{aligned}$$

	2	3	-2	-12	4
1			2	3	4
					i

$$\text{curr max} = ($$

$$A[i], \text{prev min} * A[i], \text{prev max} * A[i])$$

$$\begin{aligned}
 \text{curr max} &= 576 \\
 \text{curr min} &= -48 \\
 &4, -48, 576
 \end{aligned}$$

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         int [] A = new int[n];
10        for(int i = 0; i < n; i++){
11            A[i] = scn.nextInt();
12        }
13
14        int max = A[0];
15        int prevMax = A[0];
16        int prevMin = A[0];
17
18        for(int i = 1; i < n; i++){
19            int currMax = Math.max(A[i], Math.max(A[i] * prevMax, A[i] * prevMin));
20            int currMin = Math.min(A[i], Math.min(A[i] * prevMax, A[i] * prevMin));
21            max = Math.max(currMax, max);
22            prevMax = currMax;
23            prevMin = currMin;
24        }
25        System.out.println(max);
26    }
27 }
```