Practice Arena

Practice problems aimed to improve your coding skills.

- PRACTICE-02_SCAN-PRINT
- PRACTICE-03_TYPES
- LAB-PRAC-02_SCAN-PRINT
- LAB-PRAC-01
- PRACTICE-04_COND
- BONUS-PRAC-02
- LAB-PRAC-03_TYPES
- PRACTICE-05_COND-LOOPS
- LAB-PRAC-04_COND
- LAB-PRAC-05_CONDLOOPS
- PRACTICE-07_LOOPS-ARR
- LAB-PRAC-06_LOOPS
- LAB-PRAC-07_LOOPS-ARR
- LABEXAM-PRAC-01_MIDSEM
- PRACTICE-09_PTR-MAT
- LAB-PRAC-08_ARR-STR
- PRACTICE-10_MAT-FUN
- LAB-PRAC-09_PTR-MAT
- LAB-PRAC-10_MAT-FUN
- PRACTICE-11_FUN-PTR
- LAB-PRAC-11_FUN-PTR
- LAB-PRAC-12_FUN-STRUC
- LABEXAM-PRAC-02_ENDSEM
- LAB-PRAC-13_STRUC-NUM
- LAB-PRAC-14_SORT-MISC
  - Predecessor and Successor
  - Insertion Sort
  - Link a List
  - The United Sums of Arrays
  - Bubble Sort
  - Pretty Queues Revisited
  - Just About Sorted
  - Brick Sort
  - All My Descendants
  - Mr C likes a Majority
  - Cocktail Sort
  - All My Descendants - Part II

Mr C likes a Majority

LAB-PRAC-14_SORT-MISC

Mr C likes a Majority [20 marks]

Problem Statement
In the first line of the input we will give you n, a strictly positive odd number. In the second line we will give you n integers separated by a space. The numbers will be sorted in non-decreasing order. Store these numbers in an array called arr.

In the first line of the output, you have to tell us if there is any number in the array that appears a majority number of times i.e. ceil(n/2) times or more (recall that n will always be odd). If that is the case, print "YES" (without quotes) in the first line of the output. In the second line of the output, print the number of times the majority number appears in the array. If no number appears a majority number of times, print "NO" (without quotes) in the first line of the output, followed by the number of times the first element (i.e. arr[0]) appears in the array in the second line.

You would see that both the above problems are very easy to solve using O(n) time. The challenge is to solve these using O(log n) time using techniques we learnt in binary search (i.e. divide and conquer).

Caution

1. n may be any strictly positive odd number, even 1.
2. The numbers we give you may be negative, positive, or zero. The same number may repeat in the array. However, the array will always be sorted in non-decreasing order.
3. There should be no spaces in your output. There are two lines in your output.

EXAMPLE 1:
INPUT
7
1 1 2 2 3 3 3

OUTPUT:
NO
2

Explanation: Since n = 7, an element needs to occur 4 or more times to be in majority (ceil(7/2) = 4). No such element exists. The first element of the array appears 2 times.

EXAMPLE 2:
INPUT
9
1 3 4 4 4 4 4 4 8

OUTPUT:
YES
6

Explanation: Since n = 9, an element needs to occur 5 or more times to be in majority (ceil(9/2) = 5). The element 4 does appear 6 times.

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

Start Solving! (/editor/practice/6294)