# 3. Sequence of Stock Indicators



An investment consulting firm recommends high-performing stocks to its users. The firm assigns an indicator to each stock and ranks the stocks based on this indicator. To sort these stocks a modified version of the quicksort algorithm is used.

This variation of the quicksort algorithm takes an array of integers as an argument and works in the following way:

- If the array has fewer than 3 elements, it sorts the array using some non-recursive method.
- Otherwise, it selects *pivot* as a median of the first, the middle, and the last elements of the array. More specifically, if the array has *m* elements, it selects the *pivot* as the median of three elements with indices *0,* ⌊*m/2*⌋*, m-1* in the array, (where ⌊*m/2*⌋ denotes floor of the division of *m* by *2,* for example, ⌊5/2⌋ = 2 ). It partitions the array into two subarrays called left and right with the selected *pivot* element. The left array contains all the elements less than *pivot* in the relative order they appeared before selecting the pivot, while the right contains all the elements larger than the *pivot,* also in the order they appeared before selecting the *pivot*. Next, two recursive calls are made, one with the left array as the argument, and another with the right array as the argument.

Determine the sequence of pivot elements chosen by the algorithm.

**Example**
*stockIndicator = [8, 4, 3, 1]*

In this example, there are *n = 4* stocks in the array, so the algorithm selects *pivot* as the median of the first, the third, and the last elements which are *8, 3*, and *1* respectively: *pivot = 3.* Then recursive calls of the function are made with arrays *left = [1]* and *right = [4, 8]*. Neither of these calls produces a pivot since the arrays have fewer than 3 elements. The returned sequence of *pivot* elements is *[3]*.

**Function Description**
Complete the function *modifiedQuickSortIndicators* in the editor below.

The function has the following parameter(s):
*int stockIndicator[n]*: an array of *n* distinct integers

Returns
*int[]*: the sequence of pivot elements chosen by the algorithm.

**Constraints**

- $3 \le n \le 10^5$
- $1 \le stockIndicator[i] \le 10^9$
- All elements of a are chosen at random without replacement from the range $[1, 10^9]$

▶ **Input Format Format for Custom Testing**

▼ **Sample Case 0**

**Sample Input**

```
STDIN    Function
-----    --------
3    ->  stockIndicator[] size n = 3
3    ->  stockIndicator = [3, 1, 2]
1
2
```

**Sample Output**

```
2
```

**Explanation**
Initially, there are *3* stocks. The algorithm selects *pivot* as the median of the first, the middle, and the last elements which are *3, 1*, and *2* respectively, so *pivot = 2*. Then it recursively calls itself with arrays *left = [1]* and *right = [3]*. Neither of these calls produces a new pivot.