Problem Link:

https://leetcode.com/problems/find-closest-number-to-zero/

Problem Description:

Given an integer array `nums` of size `n`, return *the number with the value* ***closest*** *to* `0` *in* `nums`. If there are multiple answers, return *the number with the* ***largest*** *value*.

Sample Test Case:

Input: nums = [-4,-2,1,4,8]

Output: 1

Problem Approach:

Two pointers.

Solution:

We keep track of the biggest negative number, and the smallest positive number, which would logically be closest to 0. In case we find zero in the array, we return 0. As it doesn't make any sense to iterate through the array any further if we have zero itself. After iterating through the whole array, we check if the absolute value of the biggest negative number is smaller or the smallest positive number, as the smaller number of these would be closer to 0.

So, we return the smaller number of these two, and in case we have both these numbers equal to each other, then we return the positive one (Mentioned in the question, as we have to return the bigger original number).

Code (Python):

```
def findClosestNumber(self, nums: List[int]) -> int:
        biggestNegativeNumber = -sys.maxsize -1
        smallestPositiveNumber = sys.maxsize
        for i in range(len(nums)):
            if nums[i] == 0:
                return 0
            if nums[i] > biggestNegativeNumber and nums[i] < 0:
                biggestNegativeNumber = nums[i]
            elif nums[i] < smallestPositiveNumber and nums[i] > 0:
                smallestPositiveNumber = nums[i]
        if smallestPositiveNumber < (biggestNegativeNumber * -1):
            return smallestPositiveNumber
        elif (biggestNegativeNumber * -1) < smallestPositiveNumber:
            return biggestNegativeNumber
        else:
            return smallestPositiveNumber
```