

# 1005. Maximize Sum Of Array After K Negations

Solved

Easy Topics Companies

Given an integer array `nums` and an integer `k`, modify the array in the following way:

- choose an index `i` and replace `nums[i]` with `-nums[i]`.

You should apply this process exactly `k` times. You may choose the same index `i` multiple times.

Return the largest possible sum of the array after modifying it in this way.

## Example 1:

**Input:** `nums = [4,2,3], k = 1`

**Output:** 5

**Explanation:** Choose index 1 and `nums` becomes `[4,-2,3]`.

## Example 2:

**Input:** `nums = [3,-1,0,2], k = 3`

**Output:** 6

**Explanation:** Choose indices (1, 2, 2) and `nums` becomes `[3,1,0,2]`.

## Example 3:

**Input:** `nums = [2,-3,-1,5,-4], k = 2`

**Output:** 13

**Explanation:** Choose indices (1, 4) and `nums` becomes `[2,3,-1,5,4]`.

Problem List

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted

ZhukovIlya submitted at Jul 03, 2024 00:58

Runtime: 73 ms | Beats: 61.36%

Memory: 42.34 MB | Beats: 29.55%

Code: C#

```
public class Solution {
    public int LargestSumAfterKNegations(int[] nums, int k) {
        // Сортируем массив
        Array.Sort(nums);
        // Перебираем элементы массива и меняем знак
        for (int i = 0; i < nums.Length && k > 0; i++) {
            if (nums[i] < 0) {
                nums[i] = -nums[i];
                k--;
            }
        }
    }
}
```

Code

```
1 public class Solution {
2     public int LargestSumAfterKNegations(int[] nums, int k) {
3         // Сортируем массив
4         Array.Sort(nums);
5
6         // Перебираем элементы массива и меняем знак
7         for (int i = 0; i < nums.Length && k > 0; i++) {
8             if (nums[i] < 0) {
9                 nums[i] = -nums[i];
10                k--;
11            }
12        }
13    }
14 }
```

Testcase | Test Result

Accepted Runtime: 65 ms

Case 1 Case 2 Case 3

Input

nums =

[4,2,3]

k =

1

Output

Код:

```
using System.Linq;
using System;

public class Solution
```

```

{
    public int LargestSumAfterKNegations(int[] nums, int k)
    {
        // Сортируем массив
        Array.Sort(nums);

        // Перебираем элементы массива и меняем знак
        for (int i = 0; i < nums.Length && k > 0; i++)
        {
            if (nums[i] < 0)
            {
                nums[i] = -nums[i];
                k--;
            }
        }

        // Если остались операции, определяем дальнейшие действия
        // Находим минимальный элемент в массиве
        int minElement = nums.Min();

        // Если k нечетное, меняем знак минимального элемента
        if (k % 2 != 0)
        {
            for (int i = 0; i < nums.Length; i++)
            {
                if (nums[i] == minElement)
                {
                    nums[i] = -nums[i];
                    break;
                }
            }
        }

        // Возвращаем сумму массива
        return nums.Sum();
    }
}

```