

## Wiggle Sort

Given an unsorted array `nums`, reorder it **in-place** such that `nums[0] = nums[2]`.

For example, given `nums = [3, 5, 2, 1, 6, 4]`, one possible answer is `[1, 6, 2, 5, 3, 4]`.

## Solution 1

```
public class Solution {
    public void wiggleSort(int[] nums) {
        for(int i=0;i<nums.length;i++)
            if(i%2==1){
                if(nums[i-1]>nums[i]) swap(nums, i);
            }else if(i!=0 && nums[i-1]<nums[i]) swap(nums, i);
        }
    public void swap(int[] nums, int i){
        int tmp=nums[i];
        nums[i]=nums[i-1];
        nums[i-1]=tmp;
    }
}
```

written by [WalkerIX](#) original link [here](#)

## Solution 2

The final sorted `nums` needs to satisfy two conditions:

1. If `i` is odd, then `nums[i] >= nums[i - 1]` ;
2. If `i` is even, then `nums[i] <= nums[i - 1]` .

The code is just to fix the orderings of `nums` that do not satisfy 1 and 2.

```
class Solution {
public:
    void wiggleSort(vector<int>& nums) {
        int n = nums.size();
        for (int i = 1; i < n; i++)
            if (((i & 1) && nums[i] < nums[i - 1]) || (!(i & 1) && nums[i] > nums[i - 1]))
                swap(nums[i], nums[i - 1]);
    }
};
```

written by [jianchao.li.fighter](#) original link [here](#)

## Solution 3

### Python

```
class Solution(object):
    def wiggleSort(self, nums):
        for i in xrange(1, len(nums)):
            if (i % 2) ^ (nums[i] > nums[i - 1]):
                nums[i], nums[i - 1] = nums[i - 1], nums[i]

# 125 / 125 test cases passed.
# Status: Accepted
# Runtime: 80 ms
```

### Java

```
public void wiggleSort(int[] nums) {
    for (int i = 1; i < nums.length; ++i) {
        if ((i % 2 == 1) != (nums[i] > nums[i - 1])) {
            int cache = nums[i];
            nums[i] = nums[i-1];
            nums[i-1] = cache;
        }
    }
}

// 125 / 125 test cases passed.
// Status: Accepted
// Runtime: 1 ms
```

written by [dietpepsi](#) original link [here](#)

From [LeetCoder](#).