324. Wiggle Sort II

QuestionEditorial Solution

- Total Accepted: **19641**
- Total Submissions: **79847**
- Difficulty: **Medium**
- Contributors: **Admin**

Given an unsorted array nums, reorder it such that nums[0] < nums[1] > nums[2] < nums[3]....

Example:

(1) Given nums = [1, 5, 1, 1, 6, 4], one possible answer is [1, 4, 1, 5, 1, 6].

(2) Given nums = [1, 3, 2, 2, 3, 1], one possible answer is [2, 3, 1, 3, 1, 2].

Note:

You may assume all input has valid answer.

Follow Up:

Can you do it in O(n) time and/or in-place with O(1) extra space?

Credits:

Special thanks to [@dietpepsi](#) for adding this problem and creating all test cases.

```java
public static void wiggleSort(int[] nums)
{
    Arrays.sort(nums);
    int len = nums.length;
    int index;
    //(zhewei) the big num index need to move
    //when the len is odd the index should be add 1;
    //make sure that a1 >= a2
    if(len%2==0) index = len/2;
    else index = len/2+1;
    int[] a1 = new int[index];
    int[] a2 = new int[len-index];
    int k=0;
    for(int i=index-1;i>=0;i--)
    {
        a1[k++] = nums[i];
    }
    k=0;
    for(int i=len-1;i>=index;i--)
    {
```

```java
            a2[k++] = nums[i];
        }
        //at last fill the nums array
        k=0;
        for(int i=0;i<=index-1 && k<len;i++,k+=2)
        {
            nums[k] = a1[i];
            if(k<len-1)
            {
                nums[k+1]=a2[i];
            }
        }
    }
}


public static void main(String[] args) {
    // TODO Auto-generated method stub
    int[] nums = {1, 5, 1, 1, 6, 4};
    wiggleSort(nums);
    for(int k=0;k<nums.length;k++) System.out.printf("%d ",
    nums[k]);
}
```

280. Wiggle Sort

Given an unsorted array nums, reorder it in-place such that nums[0] <= nums[1] >= nums[2] <= nums[3]....
For example, given nums = [3, 5, 2, 1, 6, 4], one possible answer is [1, 6, 2, 5, 3, 4].

```java
public static void wiggleSort1(int[] nums)
{
    Arrays.sort(nums);
    for(int i=2;i<nums.length;i+=2)
    {
        //swap nums[i-1] and nums[i]
        int tmp = nums[i-1];
        nums[i-1] = nums[i];
        nums[i] = tmp;
    }
}
```

1 3 2 5 4 6