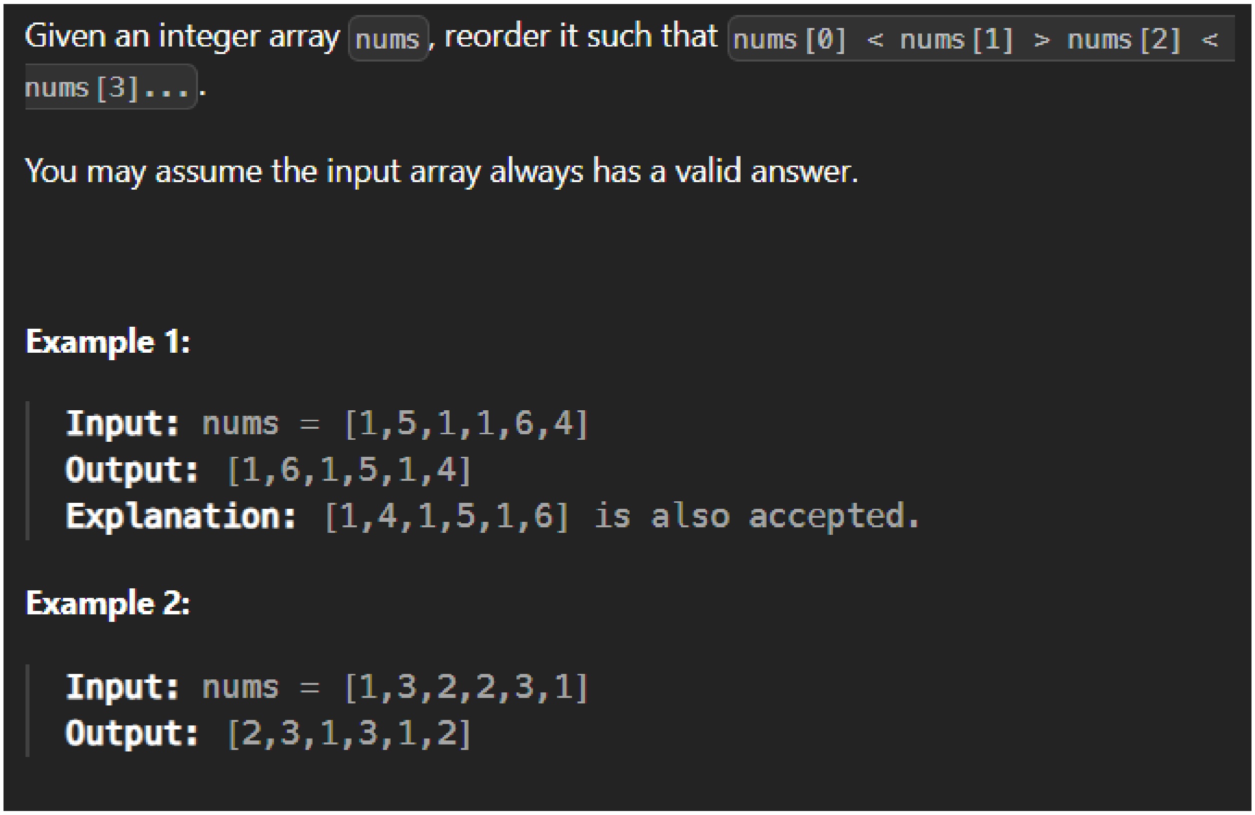
324. Wiggle Sort I (Wiggle sort 1)



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

You may assume the input array always has a valid answer.

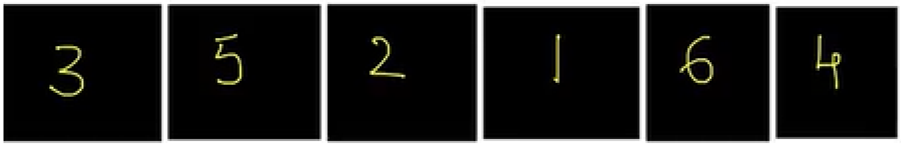
Example 1:

```
Input: nums = [1,5,1,1,6,4]
Output: [1,6,1,5,1,4]
Explanation: [1,4,1,5,1,6] is also accepted.
```

Example 2:

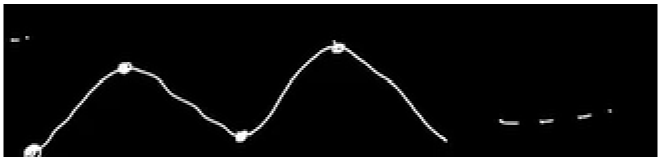
```
Input: nums = [1,3,2,2,3,1]
Output: [2,3,1,3,1,2]
```

1. Lets consider this example

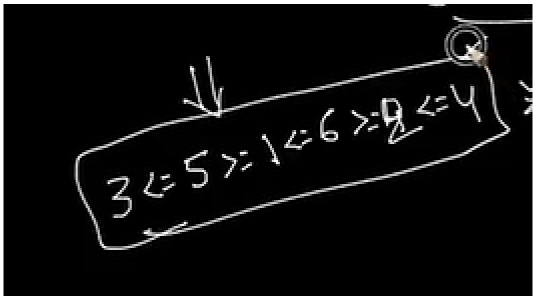


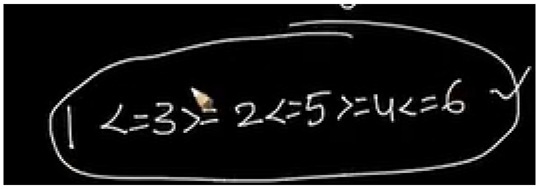
3	5	2	1	6	4

2. Here we need to sort this in WAVE like form



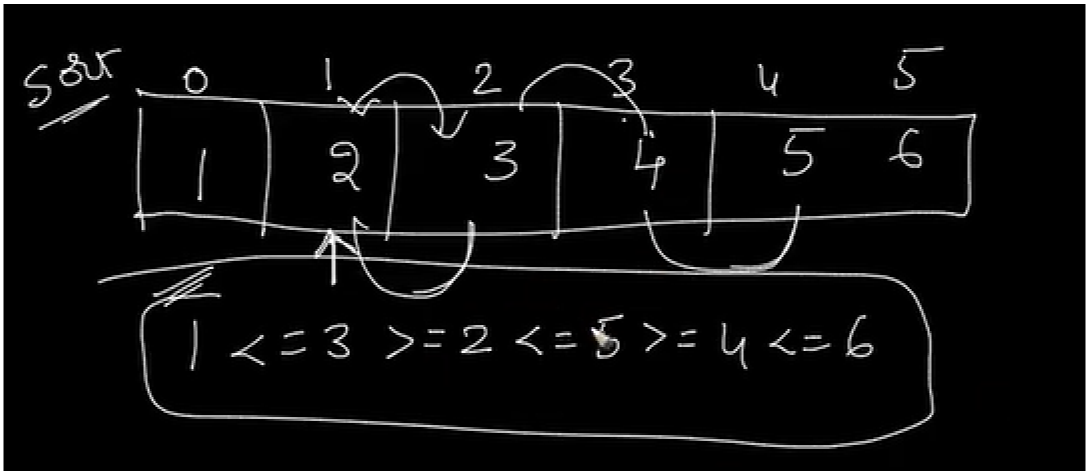
3. Few possible outputs



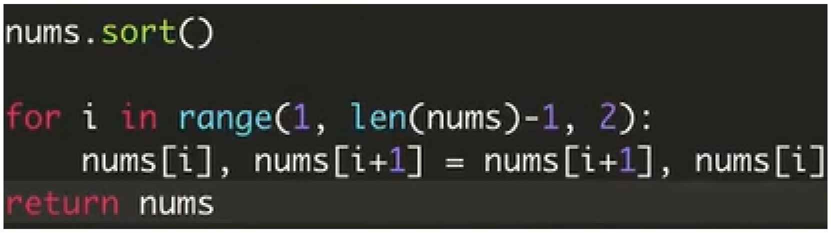


Solution using sort

1. Here first we will sort the entire array..
2. And starting from index 1 ..we’ll swap every adjacent elements



3. Python code for this approach



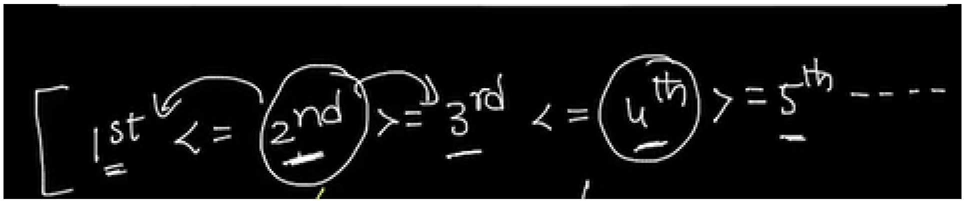
```
nums.sort()

for i in range(1, len(nums)-1, 2):
    nums[i], nums[i+1] = nums[i+1], nums[i]
return nums
```

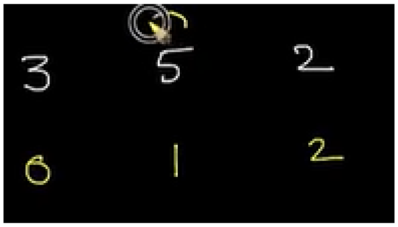
Optimized solution:

1. Let's consider the same example

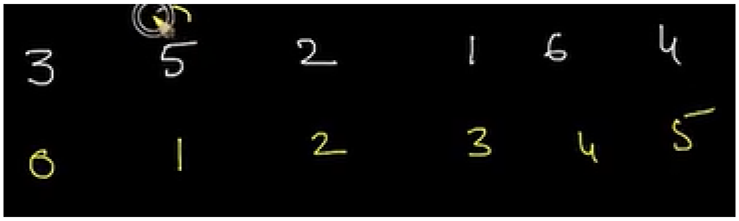
2. Here if we observe the



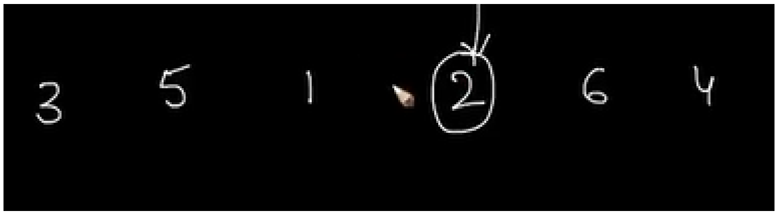
3. The even places must be greater than its neighbors
4. So we iterate our array from index 1 and compare it with i-1 & i+1..if it satisfies then we are good at index 1



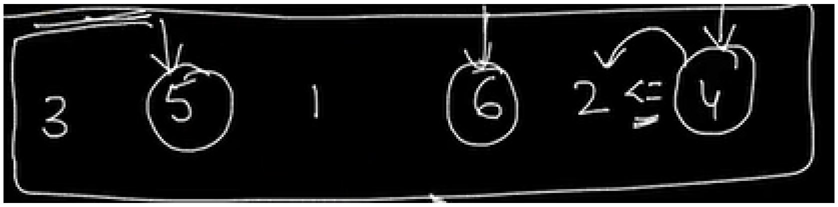
5. Next we move to index 3(here 1) and do the same thing



6. If it sd less than i-1 and we swap i and i-1 ..and next we check with i+1 ..if it is less than i+1..then we swap i and i+1 ..
7. After 1st iteration

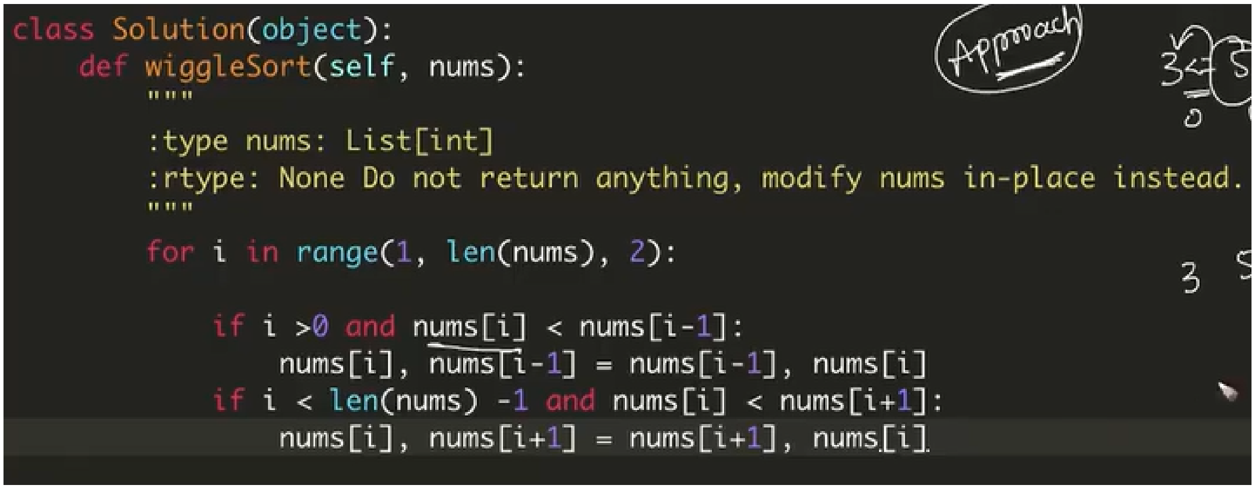


After 2nd iteration



8. This solution actually works for wiggle sort 1

9. Python code for wiggle sort 1



```python
class Solution(object):
    def wiggleSort(self, nums):
        """
        :type nums: List[int]
        :rtype: None Do not return anything, modify nums in-place instead.
        """
        for i in range(1, len(nums), 2):

            if i >0 and nums[i] < nums[i-1]:
                nums[i], nums[i-1] = nums[i-1], nums[i]
            if i < len(nums) -1 and nums[i] < nums[i+1]:
                nums[i], nums[i+1] = nums[i+1], nums[i]
```