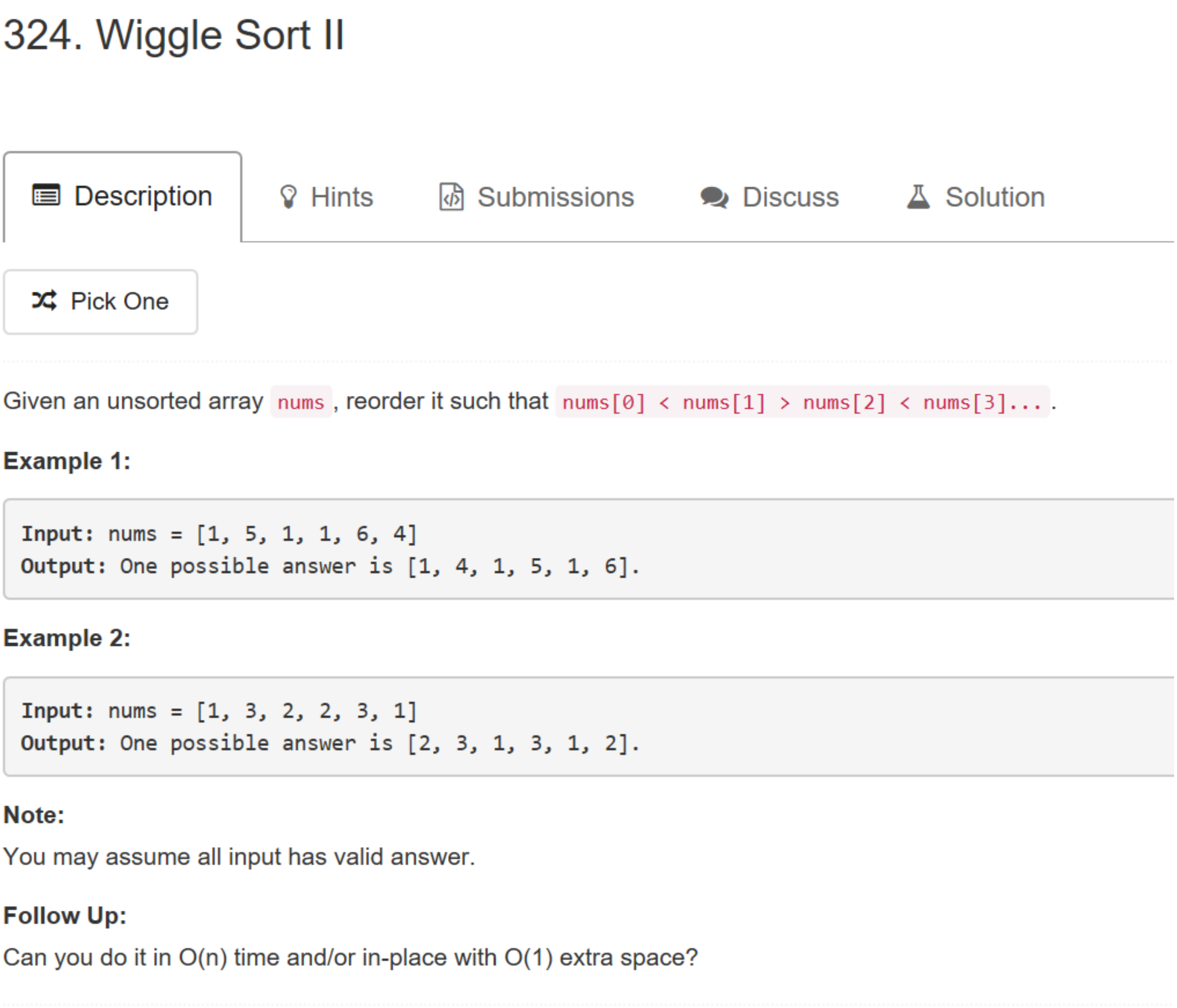


# 324. Wiggle Sort II

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

**Example 1:**

```
Input: nums = [1, 5, 1, 1, 6, 4]
Output: One possible answer is [1, 4, 1, 5, 1, 6].
```

**Example 2:**

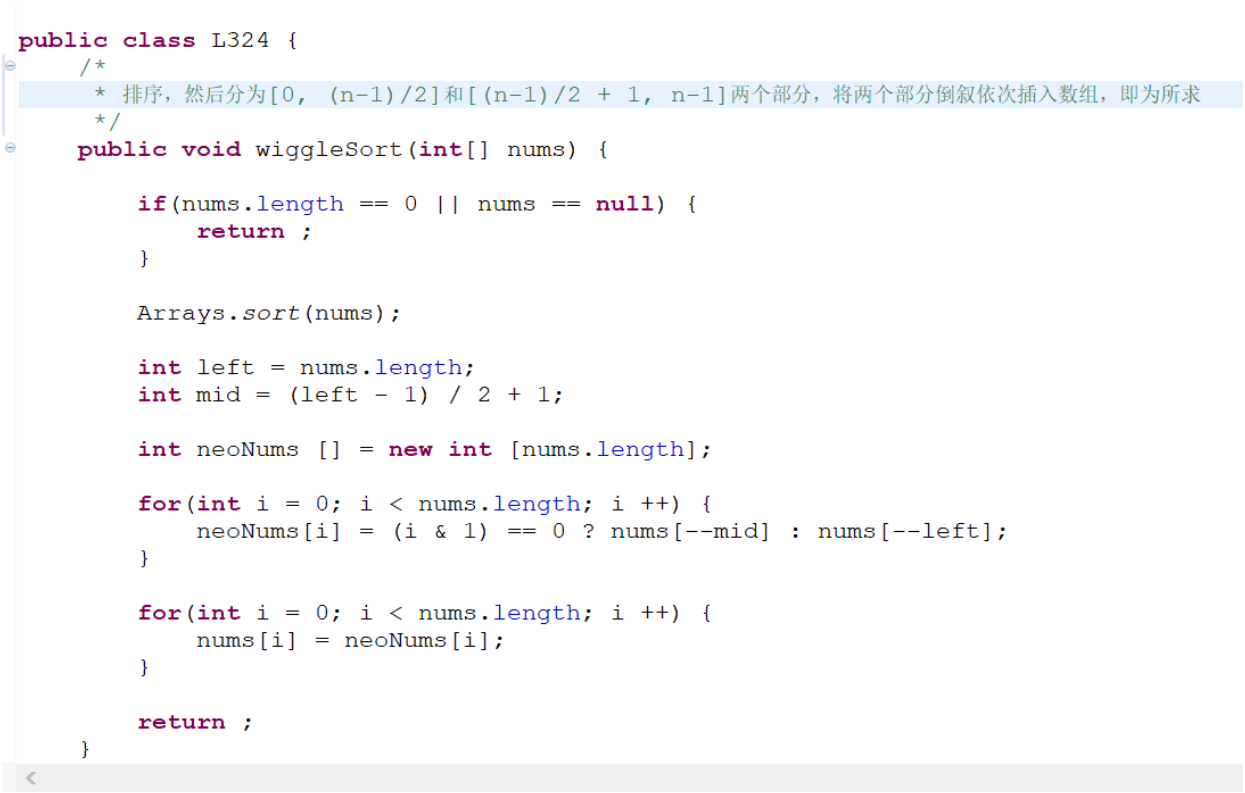
```
Input: nums = [1, 3, 2, 2, 3, 1]
Output: One possible answer is [2, 3, 1, 3, 1, 2].
```

**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?



```java
public class L324 {
    /*
     * 排序，然后分为[0， (n-1)/2]和[(n-1)/2 + 1， n-1]两个部分，将两个部分倒叙依次插入数组，即为所求
     */
    public void wiggleSort(int[] nums) {

        if(nums.length == 0 || nums == null) {
            return ;
        }

        Arrays.sort(nums);

        int left = nums.length;
        int mid = (left - 1) / 2 + 1;

        int neoNums [] = new int [nums.length];

        for(int i = 0; i < nums.length; i ++) {
            neoNums[i] = (i & 1) == 0 ? nums[--mid] : nums[--left];
        }

        for(int i = 0; i < nums.length; i ++) {
            nums[i] = neoNums[i];
        }

        return ;
    }
```