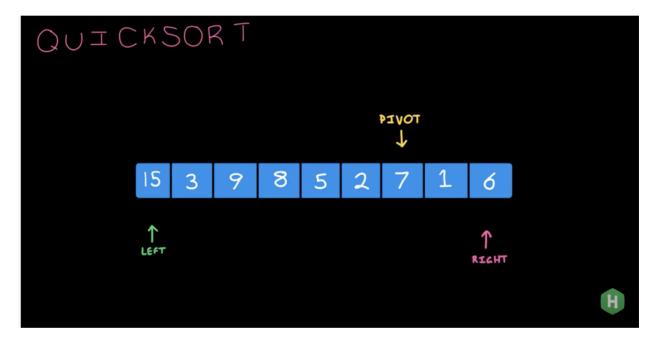
Quick Sort

步骤描述

- 1. 找到一个 pivot值
- 2. 定义 两个指针 分别从最左边和最右边开始遍历数组
- 3. 找到左边比 pivot 大的值和右边比 pivot 小的值交换
- 4. 保证最后左边的值 <=pivot, 右边的值 >=pivot
- 5. 停止条件为 left>right
- 6. 从 Left 位置把数组分成左右两个部分分别 递归 继续上面步骤

详细讲解视频及动画: https://www.youtube.com/watch?v=SLauY6PpjW4



代码实现

注意: 边界的划分十分重要。任何 '<', '>', '<=', '>=' 符号用错都可能导致无限循环 从而stackOverFlow

```
class Solution {
public int[] sortArray(int[] nums) {
    quickSort(0, nums.length-1, nums);
    return nums;
}

public void quickSort(int left, int right, int[]nums){
    int begin = left;
    int end = right;
    if(left >= right) return;
```

```
int pivot = nums[(right + left)/2];
while(left <= right) {
    while(left <= right && nums[left] < pivot) left++;
    while(left <= right && nums[right] > pivot) right--;
    if(left <= right) {
        int temp = nums[left];
        nums[left] = nums[right];
        nums[right] = temp;
        left ++;
        right --;
    }
}
quickSort(begin, right, nums);
quickSort(left, end, nums);
}</pre>
```

}