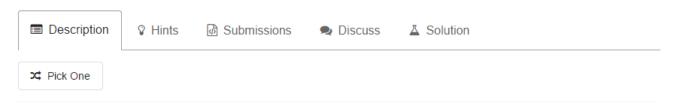
215. Kth Largest Element in an Array



Find the kth largest element in an unsorted array. Note that it is the kth largest element in the sorted order, not the kth distinct element.

Example 1:

```
Input: [3,2,1,5,6,4] and k = 2
Output: 5
```

Example 2:

```
Input: [3,2,3,1,2,4,5,5,6] and k = 4
Output: 4
```

Note:

You may assume k is always valid, $1 \le k \le array$'s length.

```
public class L215 {
       public int findKthLargest(int[] nums, int k) {
            if(nums.length == 0 || nums == null)
                  return 0;
            int left = 0;
            int right = nums.length - 1;
            while (left <= right) {
                  int location = getLocation(nums, left, right);
if(location == k - 1) {
                       break;
                  }else if (location > k - 1) {
    right = location - 1;
                 }else {
                      left = location + 1;
                }
            //记住返回的是k-1
            return nums[k-1];
        * 这儿是利用快排思想
       public int getLocation(int [] nums, int left, int right){
            int tmp = nums[left];
            int tmp_left = left;
int tmp_right = right;
//重点记住这块,多看多撸
            while(tmp_left < tmp_right) {
    while (tmp_left < tmp_right && nums[tmp_right] <= tmp) {
        tmp_right --;
    }
}</pre>
                  nums[tmp_left] = nums[tmp_right];
while (tmp_left < tmp_right && nums[tmp_left] >= tmp) {
                      tmp_left ++;
                  nums[tmp_right] = nums[tmp_left];
            nums[tmp_left] = tmp;
System.out.println(tmp_left);
return tmp_left;
       }
           public static void main(String [] args) {
   int [] array = new int [] {9,7,4,5,2,1,3,8,6,0};
   System.out.println(new L215().findKthLargest(array, 5));
}
```