/\*\*

 \* @param {number[]} nums

 \* @param {number} k

 \* @return {number}

 \*/

var findKthLargest = function(nums, k) {

    const n = nums.length;

    const quickSelect = (left, right, k) => {

        if (left === right) return nums[left];

        // 随机选择pivot（避免最坏情况）

        const pivotIndex = left + Math.floor(Math.random() \* (right - left + 1));

        const pivot = nums[pivotIndex];

        // 将pivot交换到最右边

        [nums[pivotIndex], nums[right]] = [nums[right], nums[pivotIndex]];

        let i = left;

        for (let j = left; j < right; j++) {

            // 将大于pivot的元素移到左边

            if (nums[j] > pivot) {

                [nums[i], nums[j]] = [nums[j], nums[i]];

                i++;

            }

        }

        // 将pivot放到正确位置

        [nums[i], nums[right]] = [nums[right], nums[i]];

        // 判断pivot的位置

        if (i - left + 1 > k) {

            // 在左半部分继续查找

            return quickSelect(left, i - 1, k);

        } else if (i - left + 1 < k) {

            // 在右半部分继续查找

            return quickSelect(i + 1, right, k - (i - left + 1));

        } else {

            // 找到第k大的元素

            return nums[i];

        }

    };

    return quickSelect(0, n - 1, k);

};

