Programming Test

Q1. Given an integer array nums and an integer k, return *the* k^{th} *largest element in the array*.
Note that it is the k^{th} largest element in the sorted order, not the k^{th} distinct element.
Can you solve it without sorting?

Example 1:
Input: nums = [3,2,1,5,6,4], k = 2
Output: 5
Example 2:
Input: nums = [3,2,3,1,2,4,5,5,6], k = 4
Output: 4

Q2. Suppose an array of length n sorted in ascending order is rotated between 1 and n times. For example, the array nums = [0,1,4,4,5,6,7] might become:

- **[4,5,6,7,0,1,4] if it was rotated 4 times.**
- **[0,1,4,4,5,6,7] if it was rotated 7 times.**

Notice that rotating an array [a[0], a[1], a[2], ..., a[n-1]] 1 time results in the array [a[n-1], a[0], a[1], a[2], ..., a[n-2]].
Given the sorted rotated array nums that may contain duplicates, return *the minimum element of this array*.
You must decrease the overall operation steps as much as possible.

Example 1:
Input: nums = [1,3,5]
Output: 1
Example 2:
Input: nums = [2,2,2,0,1]
Output: 0

Constraints:

- **n == nums.length**
- **1 <= n <= 5000**
- **-5000 <= nums[i] <= 5000**
- **nums is sorted and rotated between 1 and n times.**

Q3. Reverse Each Word in a Sentence but Keep Word Order.
Example:
Input → "Java is fun"
Output → "avaJ si nuf"

Q4. Check if a String is a Valid Shuffle of Two Strings.
Example:
Input → str1 = "abc", str2 = "def", result = "dabecf"
Output → Valid Shuffle

-----ALL THE BEST-----