1. Word Pattern.
   Given a pattern and a string s, find if s follows the same pattern.

   Here follow means a full match, such that there is a bijection between a letter in pattern and a non-empty word in s.
   Example 1: Input: pattern = "abba", s = "dog cat cat dog" Output: true
   Example 2: Input: pattern = "abba", s = "dog cat cat fish"
   Output: false
   Example 3: Input: pattern = "aaaa", s = "dog cat cat dog" Output: false
2. Element Appearing More Than 25% In Sorted Array.
   Given an integer array sorted in non-decreasing order, there is exactly one integer in the array that occurs more than 25% of the time, return that integer.
   Example 1: Input: arr = [1,2,2,6,6,6,6,7,10] Output: 6
   Example 2: Input: arr = [1,1] Output: 1

3. Number of Longest Increasing Subsequence.
   Given an integer array nums, return the number of longest increasing subsequences.

   Notice that the sequence has to be strictly increasing.
   Example 1:

   Input: nums = [1,3,5,4,7]
   Output: 2
   Explanation: The two longest increasing subsequences are [1, 3, 4, 7] and [1, 3, 5, 7].
   Example 2:

   Input: nums = [2,2,2,2,2]
   Output: 5
   Explanation: The length of the longest increasing subsequence is 1, and there are 5 increasing subsequences of length 1, so output 5.
4. Peak Index in a Mountain Array.
   An array arr is a mountain if the following properties hold: arr.length >= 3
   There exists some i with 0 < i < arr.length - 1 such that:
   arr[0] < arr[1] < ... < arr[i - 1] < arr[i]
   arr[i] > arr[i + 1] > ... > arr[arr.length - 1]
   Given a mountain array arr, return the index i such that arr[0] < arr[1] < ... < arr[i - 1] < arr[i] > arr[i + 1] > ... > arr[arr.length - 1].

   You must solve it in O(log(arr.length)) time complexity.
   Example 1: Input: arr = [0,1,0] Output: 1
   Example 2: Input: arr = [0,2,1,0] Output: 1
   Example 3: Input: arr = [0,10,5,2] Output: 1
5.