1. Watering Plants II

   Alice and Bob want to water n plants in their garden. The plants are arranged in a row and are labeled from 0 to n - 1 from left to right where the ith plant is located at x = i. Each plant needs a specific amount of water. Alice and Bob have a watering can each, initially full. They water the plants in the following way: Alice waters the plants in order from left to right, starting from the 0th plant. Bob waters the plants in order from right to left, starting from the (n - 1)th plant. They begin watering the plants simultaneously.

   It takes the same amount of time to water each plant regardless of how much water it needs. Alice/Bob must water the plant if they have enough in their can to fully water it. Otherwise, they first refill their can (instantaneously) then water the plant.

   In case both Alice and Bob reach the same plant, the one with more water currently in his/her watering can should water this plant. If they have the same amount of water, then Alice should water this plant.

   Given a 0-indexed integer array plants of n integers, where plants[i] is the amount of water the ith plant needs, and two integers capacityA and capacityB representing the capacities of Alice's and Bob's watering cans respectively, return the number of times they have to refill to water all the plants.

   Example 1: Input: plants = [2,2,3,3], capacityA = 5, capacityB = 5 Output: 1

   Explanation:

   - Initially, Alice and Bob have 5 units of water each in their watering cans.
   - Alice waters plant 0, Bob waters plant 3.
   - Alice and Bob now have 3 units and 2 units of water respectively.
   - Alice has enough water for plant 1, so she waters it. Bob does not have enough water for plant 2, so he refills his can then waters it.

   So, the total number of times they have to refill to water all the plants is 0 + 0 + 1 + 0 = 1.

   Example 2: Input: plants = [2,2,3,3], capacityA = 3, capacityB = 4

   Output: 2

   Explanation:

   - Initially, Alice and Bob have 3 units and 4 units of water in their watering cans respectively.
   - Alice waters plant 0, Bob waters plant 3.
   - Alice and Bob now have 1 unit of water each, and need to water plants 1 and 2 respectively.
   - Since neither of them have enough water for their current plants, they refill their cans and then water the plants.

   So, the total number of times they have to refill to water all the plants is 0 + 1 + 1 + 0 = 2.

   Example 3: Input: plants = [5], capacityA = 10, capacityB = 8

   Output: 0

   Explanation:

   - There is only one plant.
   - Alice's watering can has 10 units of water, whereas Bob's can has 8 units. Since Alice has more water in her can, she waters this plant.

   So, the total number of times they have to refill is 0.

2. Contains Duplicate II

   Given an integer array nums and an integer k, return true if there are two distinct indices i and j in the array such that nums[i] == nums[j] and abs(i - j) <= k.

   Example 1: Input: nums = [1,2,3,1], k = 3 Output: true

   Example 2: Input: nums = [1,0,1,1], k = 1 Output: true

Example 3: Input: nums = [1,2,3,1,2,3], k = 2 Output: false

3. Is Subsequence

   Given two strings s and t, return true if s is a subsequence of t, or false otherwise.

   A subsequence of a string is a new string that is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (i.e., "ace" is a subsequence of "abcde" while "aec" is not).
   Example 1: Input: s = "abc", t = "ahbgdc"
   Output: true
   Example 2: Input: s = "axc", t = "ahbgdc"
   Output: false

4. Valid Palindrome

   A phrase is a palindrome if, after converting all uppercase letters into lowercase letters and removing all non-alphanumeric characters, it reads the same forward and backward. Alphanumeric characters include letters and numbers.

   Given a string s, return true if it is a palindrome, or false otherwise.
   Example 1: Input: s = "A man, a plan, a canal: Panama"
   Output: true
   Explanation: "amanaplanacanalpanama" is a palindrome.

   Example 2: Input: s = "race a car" Output: false
   Explanation: "raceacar" is not a palindrome.

   Example 3: Input: s = " " Output: true
   Explanation: s is an empty string "" after removing non-alphanumeric characters.
   Since an empty string reads the same forward and backward, it is a palindrome.

5. Jump Game.

   You are given an integer array nums. You are initially positioned at the array's first index, and each element in the array represents your maximum jump length at that position.
   Return true if you can reach the last index, or false otherwise.

   Example 1: Input: nums = [2,3,1,1,4] Output: true

   Explanation: Jump 1 step from index 0 to 1, then 3 steps to the last index.
   Example 2: Input: nums = [3,2,1,0,4] Output: false
   Explanation: You will always arrive at index 3 no matter what. Its maximum jump length is 0, which makes it impossible to reach the last index.