Priority Queue

Assignment Questions

PW SKILLS

Q1. Given a string s, rearrange the characters of s so that any two adjacent characters are not the same.

Return any possible rearrangement of s or return "" if not possible.

Example 1:

Input: s = "aab"
Output: "aba"
Example 2:

Input: s = "aaab"
Output: ""

Q2. You are given two integer arrays nums1 and nums2 sorted in ascending order and an integer k.
Define a pair (u, v) which consists of one element from the first array and one element from the second array.
Return the k pairs (u1, v1), (u2, v2), ..., (uk, vk) with the smallest sums.

Example 1:

Input: nums1 = [1,7,11], nums2 = [2,4,6], k = 3
Output: [[1,2],[1,4],[1,6]]
Explanation: The first 3 pairs are returned from the sequence: [1,2],[1,4],[1,6],[7,2],[7,4],[11,2],[7,6],[11,4],[11,6]
Example 2:

Input: nums1 = [1,1,2], nums2 = [1,2,3], k = 2
Output: [[1,1],[1,1]]
Explanation: The first 2 pairs are returned from the sequence: [1,1],[1,1],[1,2],[2,1],[1,2],[2,2],[1,3],[1,3],[2,3]
Example 3:

Input: nums1 = [1,2], nums2 = [3], k = 3
Output: [[1,3],[2,3]]
Explanation: All possible pairs are returned from the sequence: [1,3],[2,3]

Q3. You are playing a solitaire game with three piles of stones of sizes a, b, and c respectively. Each turn you choose two different non-empty piles, take one stone from each, and add 1 point to your score. The game stops when there are fewer than two non-empty piles (meaning there are no more available moves).

Given three integers a, b, and c, return the maximum score you can get.

Example 1:

Input: a = 2, b = 4, c = 6
Output: 6
Explanation: The starting state is (2, 4, 6). One optimal set of moves is:
- Take from 1st and 3rd piles, state is now (1, 4, 5)
- Take from 1st and 3rd piles, state is now (0, 4, 4)
- Take from 2nd and 3rd piles, state is now (0, 3, 3)
- Take from 2nd and 3rd piles, state is now (0, 2, 2)

Take from 2nd and 3rd piles, state is now (0, 1, 1)
- Take from 2nd and 3rd piles, state is now (0, 0, 0)
There are fewer than two non-empty piles, so the game ends. Total: 6 points.
Example 2:

Input: a = 4, b = 4, c = 6
Output: 7
Explanation: The starting state is (4, 4, 6). One optimal set of moves is:
- Take from 1st and 2nd piles, state is now (3, 3, 6)
- Take from 1st and 3rd piles, state is now (2, 3, 5)
- Take from 1st and 3rd piles, state is now (1, 3, 4)
- Take from 1st and 3rd piles, state is now (0, 3, 3)
- Take from 2nd and 3rd piles, state is now (0, 2, 2)
- Take from 2nd and 3rd piles, state is now (0, 1, 1)
- Take from 2nd and 3rd piles, state is now (0, 0, 0)
There are fewer than two non-empty piles, so the game ends. Total: 7 points.

Q4. You are given an m x n matrix mat that has its rows sorted in non-decreasing order and an integer k.

You are allowed to choose exactly one element from each row to form an array.
Return the kth smallest array sum among all possible arrays.

Example 1:

Input: mat = [[1,3,11],[2,4,6]], k = 5
Output: 7
Explanation: Choosing one element from each row, the first k smallest sum are:
[1,2], [1,4], [3,2], [3,4], [1,6]. Where the 5th sum is 7.
Example 2:

Input: mat = [[1,3,11],[2,4,6]], k = 9
Output: 17
Example 3:

Input: mat = [[1,10,10],[1,4,5],[2,3,6]], k = 7
Output: 9
Explanation: Choosing one element from each row, the first k smallest sum are:
[1,1,2], [1,1,3], [1,4,2], [1,4,3], [1,1,6], [1,5,2], [1,5,3]. Where the 7th sum is 9.

Q5. Given that integers are read from a data stream. Find the median of elements read so for in an efficient way. For simplicity assume, there are no duplicates. For example, let us consider the streams 5, 15, 1, 3 …

After reading 1st element of stream - 5 -> median - 5
After reading 2nd element of stream - 5, 15 -> median - 10
After reading 3rd element of stream - 5, 15, 1 -> median - 5
After reading the 4th element of stream - 5, 15, 1, 3 -> median - 4, so on.