COMPUTER SCIENCE DEPT. AT COMSATS LAHORE

PRESENTS

TECHNO VERSE

EMPOWER. INNOVATE. CONNECT

VENUE: COMSATS UNIVERSITY LAHORE

DATE: 18 AND 19TH MAY 2024

@technoverse_cuilahore

@technoversecuilahore

TechnoVerse - Coding Spree: Round 1

Question 1: Tourists Pairing:

Problem Statement:

The Tourism Development Corporation of Bali needs your help! They need to calculate the minimum number of vehicles required to transport tourists to sightseeing destinations. The corporation wants to minimize operational costs by filling vehicles to capacity while ensuring all tourists have transportation.

Input:

The program will take two integer arrays as input:

1. **tourists[]:** Represents the number of tourists in each group **(size 1 to 100).**
2. **seatingCapacity[]:** Represents the seating capacities of each vehicle **(size 1 to 100).**
3. **Constraint: tourists[K] ≤ seatingCapacity[K]** for all **K** (tourists in a group cannot exceed vehicle capacity).

Output:

The program should return an integer representing the minimum number of vehicles needed to transport all tourists.

Example:

1. **Example 1:**
   - Input: [4, 4, 2, 4], [5, 5, 2, 5]
   - Output: 3
   - Explanation: 3 vehicles can efficiently transport all tourists.
2. **Example 2:**
   - Input: [2, 3, 4, 2], [2, 5, 7, 2]
   - Output: 2
   - Explanation: 2 vehicles can efficiently transport all tourists.

Test Cases:

1. **[7, 2, 3], [5, 4, 4] -> Output: 3**
2. **[1, 2, 3], [2, 2, 5] -> Output: 2**
3. **[10, 20], [15, 25] -> Output: 2**
4. **[4, 4, 2], [5, 5, 2] -> Output: 3**

Question 2: Minimum Cost to Equalize Array:

Problem Statement:

You are given an integer array **nums** and two integers cost1 and cost2. You are allowed to perform either of the following operations any number of times:

1. Choose an index **i** from **nums** and increase **nums[i]** by **1** for a cost of cost1.
2. Choose two different indices **i** and **j** from **nums** and increase **nums[i]** and nums[j] by 1 for a cost of cost2.

Return the minimum cost required to make all elements in the array equal. Since the answer may be very large, return its modulo 109 + 7.

Example:

1. **Example 1:**
   - Input: nums = [4, 1], cost1 = 5, cost2 = 2
   - Output: 15
   - Explanation: The following operations can be performed to make the values equal:
     1. Increase **nums[1]** by 1 for a cost of 5. nums becomes **[4, 2].**
     2. Increase **nums[1]** by 1 for a cost of 5. nums becomes **[4, 3].**
     3. Increase **nums[1]** by 1 for a cost of 5. nums becomes **[4, 4].**

     The total cost is 15.

2. **Example 2:**

- Input: nums = [2, 3, 3, 3, 5], cost1 = 2, cost2 = 1
- Output: 6
- Explanation: The following operations can be performed to make the values equal:

    1. Increase **nums[0]** and **nums[1]** by 1 for a cost of 1. **nums** becomes **[3, 4, 3, 3, 5]**.
    2. Increase **nums[0]** and **nums[2]** by 1 for a cost of 1. **nums** becomes **[4, 4, 4, 3, 5]**.
    3. Increase **nums[0]** and **nums[3]** by 1 for a cost of 1. **nums** becomes **[5, 4, 4, 4, 5]**.
    4. Increase **nums[1]** and **nums[2]** by 1 for a cost of 1. **nums** becomes **[5, 5, 5, 4, 5]**.
    5. Increase **nums[3]** by 1 for a cost of 2. **nums** becomes **[5, 5, 5, 5, 5]**.

  The total cost is 6.

3. **Example 3:**

- Input: nums = [3, 5, 3], cost1 = 1, cost2 = 3
- Output:
- Explanation: The following operations can be performed to make the values equal:
    1. Increase **nums[0]** by 1 for a cost of 1. **nums** becomes **[4, 5, 3].**
    2. Increase **nums[0]** by 1 for a cost of 1. **nums** becomes **[5, 5, 3]**.
    3. Increase **nums[2]** by 1 for a cost of 1. **nums** becomes **[5, 5, 4]**.
    4. Increase **nums[2]** by 1 for a cost of 1. **nums** becomes **[5, 5, 5]**.

  The total cost is 4.

Constraints:

1. **1 <= nums.length <= 10^5**
2. **1 <= nums[i] <= 10^6**
3. **1 <= cost1 <= 10^6**
4. **1 <= cost2 <= 10^6**

Question 3 Identify Sum Indices:

Problem Statement:

Given an array nums of integers and a target number target, find all possible combinations of indices from the array where the corresponding elements add up to the target number. Return a list of lists, where each inner list contains the indices of the elements that sum to the target. Each element in **nums** can be used only once in a combination.

Input:

- nums: An array of integers.
- target: An integer representing the target sum.

Output:

- A list of lists, where each inner list contains unique indices of elements in nums that sum up to target. If no such combinations exist, return an empty list [].

Constraints:

1. 1 <= nums.length <= 1000
2. 0 <= nums[i] <= 1000
3. 0 <= target <= 1000

Test Cases:

1. Input: nums = [2, 7, 11, 15], target = 9

   Output: [[0, 1]] (Indices 0 and 1 because 2 + 7 = 9)

2. Input: nums = [1, 2, 3, 4, 6], target = 10

   Output: [[1, 3, 4], [0, 1, 2, 3]] (1+3+6 =10, 1+2+3+4=10)

3. Input: nums = [2, 7, 11, 15], target = 18

   Output: [[1, 2], [0, 3]]

4. Input: nums = [3, 3, 3], target = 6

   Output: [[0, 1],[1,2],[0,2]]

5. Input: nums = [1, 2, 3, 4], target = 10

   Output: [[0,1,2,3]]

6. Input: nums = [2,4,6,8], target = 10

   Output: [[0, 4],[1,2]]

7. Input: nums = [2], target = 4

   Output: [] (Cannot repeat element in one combination)

Question 4: Largest Concatenated Number:

Problem Statement:

Given an array of strings **arr[]** of length **n** representing non-negative integers, arrange them in a manner such that, after concatenating them in order, it results in the largest possible number. Since the result may be very large, return it as a string.

Example:

Input: n = 5, **arr[]** = **{"3", "30", "34", "5", "9"}**

Output: "9534330"

Explanation: Given numbers **are {"3", "30", "34", "5", "9"},** the arrangement **"9534330"** gives the largest value.

Constraints:

1. **1 ≤ n ≤ 100**
2. **1 ≤ Length of each string in arr ≤ 100**

Test Cases:

1. **Input: n = , arr[] = {"1", "20", "23"}**

   Output: "23120"

2. **Input: n = 4, arr[] = {"54", "546", "548", "60"}**

   Output: "6054854654"

3. **Input: n = 3, arr[] = {"12", "121", "120"}**

   Output:"12121120"

GOOD LUCK