ASSIGNMENT (19.06.24)

1. Odd String Difference

You are given an array of equal-length strings words. Assume that the length of each string is n.Each string words[i] can be converted into a difference integer array difference[i] of length n - 1 where difference[i][j] = words[i][j+1] - words[i][j] where 0 <= j <= n - 2. Note that the difference between two letters is the difference between their positions in the alphabet i.e. the position of 'a' is 0, 'b' is 1, and 'z' is 25.

For example, for the string "acb", the difference integer array is [2 - 0, 1 - 2] = [2, -1].All the strings in words have the same difference integer array, except one. You should find that string.

Return *the string in* words *that has different difference integer array.*

Example 1:

Input: words = ["adc","wzy","abc"]

Output: "abc"

Explanation:

- The difference integer array of "adc" is [3 - 0, 2 - 3] = [3, -1].

- The difference integer array of "wzy" is [25 - 22, 24 - 25]= [3, -1].

- The difference integer array of "abc" is [1 - 0, 2 - 1] = [1, 1].

The odd array out is [1, 1], so we return the corresponding string, "abc".

Example 2:

Input: words = ["aaa","bob","ccc","ddd"]

Output: "bob"

Explanation: All the integer arrays are [0, 0] except for "bob", which corresponds to [13, -13].

Constraints:

- 3 <= words.length <= 100
- n == words[i].length
- 2 <= n <= 20
- words[i] consists of lowercase English letters.

2. Words Within Two Edits of Dictionary

You are given two string arrays, queries and dictionary. All words in each array comprise of lowercase English letters and have the same length.In one edit you can take a word from queries, and change any letter in it to any other letter. Find all words from queries that, after a maximum of two edits, equal some word from dictionary.

Return *a list of all words from* queries*, that match with some word from* dictionary *after a maximum of two edits*. Return the words in the same order they appear in queries.

Example 1:

Input: queries = ["word","note","ants","wood"], dictionary = ["wood","joke","moat"]

Output: ["word","note","wood"]

Explanation:

- Changing the 'r' in "word" to 'o' allows it to equal the dictionary word "wood".

- Changing the 'n' to 'j' and the 't' to 'k' in "note" changes it to "joke".

- It would take more than 2 edits for "ants" to equal a dictionary word.

- "wood" can remain unchanged (0 edits) and match the corresponding dictionary word.

Thus, we return ["word","note","wood"].

Example 2:

Input: queries = ["yes"], dictionary = ["not"]

Output: []

Explanation:

Applying any two edits to "yes" cannot make it equal to "not". Thus, we return an empty array.

Constraints:

- 1 <= queries.length, dictionary.length <= 100
- n == queries[i].length == dictionary[j].length
- 1 <= n <= 100
- All queries[i] and dictionary[j] are composed of lowercase English letters.

3. Destroy Sequential Targets

You are given a 0-indexed array nums consisting of positive integers, representing targets on a number line. You are also given an integer space.

You have a machine which can destroy targets. Seeding the machine with some nums[i] allows it to destroy all targets with values that can be represented as nums[i] + c * space, where c is any non-negative integer. You want to destroy the maximum number of targets in nums.

Return *the minimum value of* nums[i] *you can seed the machine with to destroy the maximum number of targets.*

Example 1:

Input: nums = [3,7,8,1,1,5], space = 2

Output: 1

Explanation: If we seed the machine with nums[3], then we destroy all targets equal to 1,3,5,7,9,...

In this case, we would destroy 5 total targets (all except for nums[2]).

It is impossible to destroy more than 5 targets, so we return nums[3].

Example 2:

Input: nums = [1,3,5,2,4,6], space = 2

Output: 1

Explanation: Seeding the machine with nums[0], or nums[3] destroys 3 targets.

It is not possible to destroy more than 3 targets.

Since nums[0] is the minimal integer that can destroy 3 targets, we return 1.

Example 3:

Input: nums = [6,2,5], space = 100

Output: 2

Explanation: Whatever initial seed we select, we can only destroy 1 target. The minimal seed is nums[1].

Constraints:

- 1 <= nums.length <= 105
- 1 <= nums[i] <= 109
- 1 <= space <= 109

4. Next Greater Element IV

You are given a 0-indexed array of non-negative integers nums. For each integer in nums, you must find its respective second greater integer.

The second greater integer of nums[i] is nums[j] such that:

- j > i

- nums[j] > nums[i]
- There exists exactly one index k such that nums[k] > nums[i] and i < k < j.

If there is no such nums[j], the second greater integer is considered to be -1.

- For example, in the array [1, 2, 4, 3], the second greater integer of 1 is 4, 2 is 3, and that of 3 and 4 is -1.

Return *an integer array* answer*, where* answer[i] *is the second greater integer of* nums[i]*.*

Example 1:

Input: nums = [2,4,0,9,6]

Output: [9,6,6,-1,-1]

Explanation:

0th index: 4 is the first integer greater than 2, and 9 is the second integer greater than 2, to the right of 2.

1st index: 9 is the first, and 6 is the second integer greater than 4, to the right of 4.

2nd index: 9 is the first, and 6 is the second integer greater than 0, to the right of 0.

3rd index: There is no integer greater than 9 to its right, so the second greater integer is considered to be -1.

4th index: There is no integer greater than 6 to its right, so the second greater integer is considered to be -1.

Thus, we return [9,6,6,-1,-1].

Example 2:

Input: nums = [3,3]

Output: [-1,-1]

Explanation:

We return [-1,-1] since neither integer has any integer greater than it.

Constraints:

- 1 <= nums.length <= 105
- 0 <= nums[i] <= 109

5. Average Value of Even Numbers That Are Divisible by Three

Given an integer array nums of positive integers, return *the average value of all even integers that are divisible by* 3*.*

Note that the average of n elements is the sum of the n elements divided by n and rounded down to the nearest integer.

Example 1:

Input: nums = [1,3,6,10,12,15]

Output: 9

Explanation: 6 and 12 are even numbers that are divisible by 3. (6 + 12) / 2 = 9.

Example 2:

Input: nums = [1,2,4,7,10]

Output: 0

Explanation: There is no single number that satisfies the requirement, so return 0.

Constraints:

- 1 <= nums.length <= 1000
- 1 <= nums[i] <= 1000

You are given two positive integers n and target.

An integer is considered beautiful if the sum of its digits is less than or equal to target.

Return the *minimum non-negative integer* x *such that* n + x *is beautiful*. The input will be generated such that it is always possible to make n beautiful.

Example 1:

Input: n = 16, target = 6

Output: 4

Explanation: Initially n is 16 and its digit sum is 1 + 6 = 7. After adding 4, n becomes 20 and digit sum becomes 2 + 0 = 2. It can be shown that we can not make n beautiful with adding non-negative integer less than 4.

Example 2:

Input: n = 467, target = 6

Output: 33

Explanation: Initially n is 467 and its digit sum is 4 + 6 + 7 = 17. After adding 33, n becomes 500 and digit sum becomes 5 + 0 + 0 = 5. It can be shown that we can not make n beautiful with adding non-negative integer less than 33.

Example 3:

Input: n = 1, target = 1

Output: 0

Explanation: Initially n is 1 and its digit sum is 1, which is already smaller than or equal to target.

Constraints:

- 1 <= n <= 1012
- 1 <= target <= 150
- The input will be generated such that it is always possible to make n beautiful.

6. Most Popular Video Creator

You are given two string arrays creators and ids, and an integer array views, all of length n. The ith video on a platform was created by creator[i], has an id of ids[i], and has views[i] views.

The popularity of a creator is the sum of the number of views on all of the creator's videos. Find the creator with the highest popularity and the id of their most viewed video.

- If multiple creators have the highest popularity, find all of them.
- If multiple videos have the highest view count for a creator, find the lexicographically smallest id.

Return *a 2D array of strings* answer *where* answer[i] = [creatori, idi] *means that* creatori *has the highest popularity and* idi *is the id of their most popular video.* The answer can be returned in any order.

Example 1:

Input: creators = ["alice","bob","alice","chris"], ids = ["one","two","three","four"], views = [5,10,5,4]

Output: [["alice","one"],["bob","two"]]

Explanation:

The popularity of alice is 5 + 5 = 10.

The popularity of bob is 10.

The popularity of chris is 4.

alice and bob are the most popular creators.

For bob, the video with the highest view count is "two".

For alice, the videos with the highest view count are "one" and "three". Since "one" is lexicographically smaller than "three", it is included in the answer.

Example 2:

Input: creators = ["alice","alice","alice"], ids = ["a","b","c"], views = [1,2,2]

Output: [["alice","b"]]

Explanation:

The videos with id "b" and "c" have the highest view count.

Since "b" is lexicographically smaller than "c", it is included in the answer.

Constraints:

- n == creators.length == ids.length == views.length
- 1 <= n <= 105
- 1 <= creators[i].length, ids[i].length <= 5
- creators[i] and ids[i] consist only of lowercase English letters.
- 0 <= views[i] <= 105

7. Minimum Addition to Make Integer Beautiful
You are given two positive integers n and target.

An integer is considered beautiful if the sum of its digits is less than or equal to target.

Return the *minimum non-negative integer* x *such that* n + x *is beautiful*. The input will be generated such that it is always possible to make n beautiful.

Example 1:

Input: n = 16, target = 6
Output: 4
Explanation: Initially n is 16 and its digit sum is 1 + 6 = 7. After adding 4, n becomes 20 and digit sum becomes 2 + 0 = 2. It can be shown that we can not make n beautiful with adding non-negative integer less than 4.

Example 2:

Input: n = 467, target = 6
Output: 33
Explanation: Initially n is 467 and its digit sum is 4 + 6 + 7 = 17. After adding 33, n becomes 500 and digit sum becomes 5 + 0 + 0 = 5. It can be shown that we can not make n beautiful with adding non-negative integer less than 33.

Example 3:

Input: n = 1, target = 1
Output: 0
Explanation: Initially n is 1 and its digit sum is 1, which is already smaller than or equal to target.

Constraints:

- 1 <= n <= 1012
- 1 <= target <= 150
- The input will be generated such that it is always possible to make n beautiful.

8. Split Message Based on Limit
You are given a string, message, and a positive integer, limit.

You must split message into one or more parts based on limit. Each resulting part should have the suffix "<a/b>", where "b" is to be replaced with the total number of parts and "a" is to be replaced with the index of the part, starting from 1 and going up to b. Additionally, the length of each resulting part (including its suffix) should be equal to limit, except for the last part whose length can be at most limit.

The resulting parts should be formed such that when their suffixes are removed and they are all concatenated in order, they should be equal to message. Also, the result should contain as few parts as possible.

Return *the parts* message *would be split into as an array of strings*. If it is impossible to split message as required, return *an empty array*.

Example 1:

Input: message = "this is really a very awesome message", limit = 9
Output: ["thi<1/14>","s i<2/14>","s r<3/14>","eal<4/14>","ly <5/14>","a v<6/14>","ery<7/14>"," aw<8/14>","eso<9/14>","me<10/14>"," m<11/14>","es<12/14>","sa<13/14>","ge<14/14>"]
Explanation:
The first 9 parts take 3 characters each from the beginning of message.
The next 5 parts take 2 characters each to finish splitting message.
In this example, each part, including the last, has length 9.
It can be shown it is not possible to split message into less than 14 parts.

Example 2:

Input: message = "short message", limit = 15
Output: ["short mess<1/2>","age<2/2>"]
Explanation:
Under the given constraints, the string can be split into two parts:
- The first part comprises of the first 10 characters, and has a length 15.
- The next part comprises of the last 3 characters, and has a length 8.

Constraints:

- 1 <= message.length <= 104
- message consists only of lowercase English letters and ' '.
- 1 <= limit <= 104