

BASIC PROGRAMMING

April 27

Given two version numbers, `version1` and `version2`, compare them.

Version numbers consist of **one or more revisions** joined by a dot '.'. Each revision consists of **digits** and may contain leading **zeros**. Every revision contains **at least one character**. Revisions are **0-indexed from left to right**, with the leftmost revision being revision 0, the next revision being revision 1, and so on. For example 2.5.33 and 0.1 are valid version numbers.

To compare version numbers, compare their revisions in **left-to-right order**. Revisions are compared using their **integer value ignoring any leading zeros**. This means that revisions 1 and 001 are considered **equal**. If a version number does not specify a revision at an index, then **treat the revision as 0**. For example, version 1.0 is less than version 1.1 because their revision 0s are the same, but their revision 1s are 0 and 1 respectively, and $0 < 1$.

Return the following:

- If `version1 < version2`, return -1.
- If `version1 > version2`, return 1.
- Otherwise, return 0.

Example 1:

Input: `version1 = "1.01"`, `version2 = "1.001"`

Output: 0

Explanation: Ignoring leading zeroes, both "01" and "001" represent the same integer "1".

Example 2:

Input: `version1 = "1.0"`, `version2 = "1.0.0"`

Output: 0

Explanation: `version1` does not specify revision 2, which means it is treated as "0".

Example 3:

Input: `version1 = "0.1"`, `version2 = "1.1"`

Output: -1

Explanation: `version1`'s revision 0 is "0", while `version2`'s revision 0 is "1". $0 < 1$, so `version1 < version2`.

problem 1:

given a string S and rotation R and an integer n

you need to rotate the string for n rotations L(left rotation), R(right rotation)

sample input:

s="ZOHOCORPORATION"

r="L-ROTATION"

n=4

output:"CORPORATIONZOH

SAMPLE INPUT:

s="HELLO"

r="R-ROTATION"

n=5

OUTPUT: "ELLOH"

PROBLEM 2:

patter printing:

matrix = ["abc","def","ghi"]

sample input1: 4

output:

abc

def

ghi

sample input2: 5

abc

def

ghi

PROBLEM 3:

given a string, find the maximum distance between the same two characters

sample input1: abacdeefghdei

sample output: 5

sample input: abcdba

sample output: 4

problem 4:

given a string. find the longest continuous substring without repeated characters

sample input: aweeacwerjkl

output: acwerjkl

sample input: asdsdah

output: sdah

Problem 5:

given a array and integer N. Find the number N is a factorial of the array. return true if it is a factorial else return false

[2,3,5]

sample input:

N = 12

output: true

sample input:

N = 14

output: False

problem 6:

given a string. find the distance between two same characters and return the maximum distance

sample input: "abcdefeb"

sample output: 5

sample input: "wwwxcvcxw"

output: 5

Given an encoded string, return its decoded string.

The encoding rule is: $k[\text{encoded_string}]$, where the `encoded_string` inside the square brackets is being repeated exactly k times. Note that k is guaranteed to be a positive integer.

You may assume that the input string is always valid; there are no extra white spaces, square brackets are well-formed, etc. Furthermore, you may assume that the original data does not contain any digits and that digits are only for those repeat numbers, k . For example, there will not be input like `3a` or `2[4]`. The test cases are generated so that the length of the output will never exceed 10^5 .

Example 1:

Input: `s = "3[a]2[bc]"`

Output: "aaabcbc"

Example 2:

Input: `s = "3[a2[c]]"`

Output: "accaccacc"

Example 3:

Input: `s = "2[abc]3[cd]ef"`

Output: "abcabccdcdef"

Group Anagrams

Description

Given an array of strings `strs`, group the anagrams together. You can return the answer in any order.

An Anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

Example 1:

Input: `strs = ["eat", "tea", "tan", "ate", "nat", "bat"]`

Output: `[["bat"], ["nat", "tan"], ["ate", "eat", "tea"]]`

Example 2:

Input: `strs = [""]`

Output: `[[""]]`

Example 3:

Input: `strs = ["a"]`

Output: `[["a"]]`

Longest Common Prefix

Description

Write a function to find the longest common prefix string amongst an array of strings.

If there is no common prefix, return an empty string "".

Example 1:

Input: strs = ["flower", "flow", "flight"]
Output: "fl"

Example 2:

Input: strs = ["dog", "racecar", "car"]
Output: ""

Explanation: There is no common prefix among the input strings.

Best Time to Buy and Sell Stock

Description

You are given an array prices where $\text{prices}[i]$ is the price of a given stock on the i^{th} day.

You want to maximize your profit by choosing a **single day** to buy one stock and choosing a **different day in the future** to sell that stock.

Return *the maximum profit you can achieve from this transaction*. If you cannot achieve any profit, return 0.

Example 1:

Input: prices = [7,1,5,3,6,4]
Output: 5

Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = $6 - 1 = 5$. Note that buying on day 2 and selling on day 1 is not allowed because you must buy before you sell.

Example 2:

Input: prices = [7,6,4,3,1]
Output: 0
Explanation: In this case, no transactions are done and the max profit

1.). You are given a 0-indexed integer array buses of length n, where buses[i] represents the departure time of the ith bus. You are also given a 0-indexed integer array passengers of length m, where passengers[j] represents the arrival time of the jth passenger. All bus departure times are unique. All passenger arrival times are unique. You are given an integer capacity, which represents the maximum number of passengers that can get on each bus.

When a passenger arrives, they will wait in line for the next available bus. You can get on a bus that departs at x minutes if you arrive at y minutes where $y \leq x$ and the bus is not full. Passengers with the earliest arrival times get on the bus first.

More formally when a bus arrives, either:

If capacity or fewer passengers are waiting for a bus, they will all get on the bus, or

The capacity passengers with the earliest arrival times will get on the bus.

Return the latest time you may arrive at the bus station to catch a bus. You cannot arrive at the same time as another passenger.

Note: The arrays buses and passengers are not necessarily sorted.

Example 1:

Input: buses= [10, 20] passengers= [2, 17, 18, 19], capacity= 2

Output: 16

Explanation: Suppose you arrive at time 16.

At time 10, the first bus departs with the 0th passenger.

At time 20, the second bus departs with you and the 1st passenger.

Note that you may not arrive at the same time as another passenger, which is why you must arrive before the 1st passenger to catch the bus.

Example 2:

Input: buses = [20, 30, 10], passengers s = [19, 13, 26, 4, 25, 11, 21] capacity= 2

Output: 20

Explanation: Suppose you arrive at time 20.

At time 10, the first bus departs with the 3rd passenger.

At time 20, the second bus departs with the 5th and 1st passengers.

At time 30, the third bus departs with the 0th passenger and you.

Notice if you had arrived any later, then the 6th passenger would have taken your seat on the third bus.

4, 11, 13, 19, 21, 25, 26

10-4

20 - 11, 13

30 - 19, 21

2.) Given an array A[], write a function that segregates even and odd numbers, The function should put all even numbers first and then odd numbers, arrange them in descending order respectively. Should not use an extra space

input= \{12, 34, 45, 9, 8, 90, 3\}

output= \{90, 34, 12, 8, 45, 9, 3\}

3.) Design a hit counter which counts the number of hits received in the past 5 minutes (i.e., the past 300 seconds).

Your system should accept a timestamp parameter (in seconds granularity), and you may assume that calls are being made to the system in chronological order (i.e., timestamp is monotonically increasing). Several hits may arrive roughly at the same time.

Implement the HitCounter class:

HitCounter() Initializes the object of the hit counter system.

void hit(int timestamp) Records a hit that happened at timestamp (in seconds). Several hits may happen at the same timestamp.
int getHits(int timestamp) Returns the number of hits in the past 5 minutes from timestamp (i.e., the past 300 seconds).

Example 1:

Input

["HitCounter", "hit", "hit", "hit", "getHits", "hit", "getHits", "getHits"]

[[], [1], [2], [3] [4], [300], [300], [301]]

Output

[null, null, null, null, 3, null, 4, 3]

Explanation

```
HitCounter hitCounter = new HitCounter();

hitCounter.hit(1);

// hit at timestamp 1.

hitCounter.hit(2); // hit at timestamp 2.

hitCounter.hit(3); // hit at timestamp 3.

hitCounter.getHits(4); // get hits at timestamp 4, return 3.

hitCounter.hit(300); // hit at timestamp 300.

hitCounter.getHits(300); // get hits at timestamp 300, return 4.

hitCounter.getHits(301); // get hits at timestamp 301, return 3.
```

4.) There is one meeting room in a firm. There are N meetings in the form of $(S[i], F[i])$ where $S[i]$ is the start time of meeting i and $F[i]$ is the finish time of meeting i. The task is to find the maximum number of meetings that can be accommodated in the meeting room. Print all meeting numbers

Answer:

Input: $s[] = \{1, 3, 0, 5, 8, 5\}$, $f[] = \{2, 4, 6, 7, 9, 9\}$

Output: 1245

First meeting [1, 2]

Second meeting [3, 4]

fourth meeting [5, 7]

Fifth meeting [8, 9]

5.) Write a program to implement fill-with-color that we come across on image editing programs like Coral Draw, MS Paint, etc., An $N \times N$ 2D matrix contains set of distribution of different colours. We call a function `fillWithColor`(equivalent to clicking on a colour) on a cell, say "red" with a new colour say "blue", where the colour distribution bleeds outward (look for top, left, right and bottom indices to simulate bleeding out) in a pixel by pixel manner until a cell that is not "red" is encountered.

Example:

I/P

N: 7

Enter color to fill: BI

Enter cell to fillWithColor: 3,3

O/P

Y BI BI BI BI BI Y

YYBIBIBIYY

YY BIBI BIYY

Y BI BI BI BI BI Y

=====

5/5/24

PROBLEM 1

Given an array, the distance between two array values is the number of indices between them. Find the minimum distance between any pair of equal elements in the array. If no such value exists, return -1 .

Sample Input 1

6

713417

Sample Output 1

3

Sample Input 2

5

123410

Sample Output 2

-1

Problem 2

There is a large pile of socks that must be paired by color. Given an array of integers representing the color of each sock, determine how many pairs of socks with matching colors there are.

Sample Input

10 20 20 10 10 30 50 10 20

Sample Output

3

Problem 3

Given a rational number, print its corresponding equivalent mixed fraction representation.

Note: Safely assume that float's precision in $3 / C + 1$ is enough to handle the given inputs

Sample Input 1

2.5

Sample Output 1

2 1/2

Sample Input 2

6.25

Sample Output 2

6 1/4

Problem 4

There is a string S , of lowercase English letters, that is repeated infinitely many times. Given an integer N , find and print the number of letter a's in the first N letters of the infinite string.

Note: You should solve this without using any extra memory

Sample Input 1

Aba

10

Sample Output 1

7

Sample Input 2

abacac

10

Sample Output

4

Problem 5

Find if a string A can be formed by removing a few characters from the other given string B. If so String A becomes a sub-sequence of String B. A is said to be a sub-sequence of B, if A can be obtained by deleting some characters of B without changing the order of the remaining characters in B. Determine if A is a sub-sequence of B.

Sample Input 1:

Set Sangeet

Sample Output 2: YES

Sample Input 2:

Zoho India

Sample Output 2: NO

Sample Input 3:

Set Step

Sample Output 3: NO

Problem 6

There is an array of random integers. This array is passed to a class to which you have no access. The class has a method ConstructBST which takes the array as input and constructs a BST and stores it inside the class object. Remember, you do not have access to the constructed tree as well. All you have is the array which was initially passed to the class. Using this array, you have to print the in-order traversal of the constructed binary search tree.

Note: Print the in-order traversal of the BST without constructing the BST

Problem 7

You have three stacks of cylinders where each cylinder has the same diameter, but they may vary in height. You can change the height of a stack by removing and discarding its topmost cylinder any number of times.

Find the maximum possible height of the stacks, such that all of the stacks are the same height. This means you must remove zero or more cylinders from the top of zero or more of the three stacks until they are all the same height, then print the height. The removals must be performed in such a way as to maximize the height.

Note: An empty stack is still a stack.

Input Format

The first line contains three space-separated integers n_1, n_2, n_3 describing the respective number of cylinders in stacks 1, 2, 3. The subsequent lines describe the respective heights of each cylinder in a stack from top to bottom:

The second line contains n_1 space-separated integers describing the cylinder heights in the stack.

The third line contains n_2 space-separated integers describing the cylinder heights in the stack.

The fourth line contains n_3 space-separated integers describing the cylinder heights in the stack.

Sample Input 1:

534

32111

432

1141

Sample Output 1: 5

Feb 13 / 2024

Problem 1:

Print the following pattern

Input:

5

Output:

1

2 6

3 7 10

4 8 11 13

5 9 12 14 15

Problem 2.

Write a program to print all the LEADERS in the array. An element is leader if it is greater than all the elements to its right side. And the rightmost element is always a leader.

Solve it in O(N)

Input 1:

Arr[] = \{7, 8, 3, 4, 6, 2\}

Output 1: 862

Input 2:

Arr[] = \{5, 5, 3, 2, 1\}

Output 2:

5321

Problem 3:

Check if the frequency of all characters in the given string are equal when you remove exactly one character from the string, solve it in O(N)

Example 1:

Input: cdcddca

Output: true

Explanation: if "a" is removed, "c and d ^ prime prime will have the same frequency of 3.

Example 2:

Input: aaabbbcc

Output: false

Explanation: Frequency of the characters in the given string wont be equal if any 1 char is removed.

Problem 4:

Given an array of integers find the sum of the values of the longest sub array containing at most two distinct numbers. Give an optimised solution.

Input: [155155721665]

Output: 22

Note: In this input the corresponding sub array(155155)

Problem 5:

Given an array of integers 'nums' and an integer 'target', find three integers in nums such that the sum is closest to target. Return the sum of those three integers. Assume that there is only one valid answer.

Input:

nums= [-1, 2, 1, -4], target= 1

Output: 2

Explanation: The sum that is closest to the target is 2 (-1 + 2 + 1 = 2)