WEEK 1 PROBLEMS

Problem 1 : Not Quite Lisp

Santa was hoping for a white Christmas, but his weather machine's "snow" function is powered by stars, and he's fresh out! To save Christmas, he needs you to collect fifty stars by December 25th.

Collect stars by helping Santa solve puzzles. Two puzzles will be made available on each day in the Advent calendar; the second puzzle is unlocked when you complete the first. Each puzzle grants one star. Good luck!

Here's an easy puzzle to warm you up.

Santa is trying to deliver presents in a large apartment building, but he can't find the right floor - the directions he got are a little confusing. He starts on the ground floor (floor 0) and then follows the instructions one character at a time.

An opening parenthesis, (, means he should go up one floor, and a closing parenthesis,), means he should go down one floor.

The apartment building is very tall, and the basement is very deep; he will never find the top or bottom floors.

For example:

(()) and ()() both result in floor 0.
(((and (()(()(both result in floor 3.
))(((((also results in floor 3.
()) and))(both result in floor -1 (the first basement level).
))) and)())()) both result in floor -3.

To what floor do the instructions take Santa?

Problem 2 : No Time for a Taxicab

Santa's sleigh uses a very high-precision clock to guide its movements, and the clock's oscillator is regulated by stars. Unfortunately, the stars have been stolen... by the Easter Bunny. To save Christmas, Santa needs you to retrieve all fifty stars by December 25th.

Collect stars by solving puzzles. Two puzzles will be made available on each day in the Advent calendar; the second puzzle is unlocked when you complete the first. Each puzzle grants one star. Good luck!

You're airdropped near Easter Bunny Headquarters in a city somewhere. "Near", unfortunately, is as close as you can get - the instructions on the Easter Bunny Recruiting Document the Elves intercepted start here, and nobody had time to work them out further.

The Document indicates that you should start at the given coordinates (where you just landed) and face North. Then, follow the provided sequence: either turn left (L) or right (R) 90 degrees, then walk forward the given number of blocks, ending at a new intersection.

There's no time to follow such ridiculous instructions on foot, though, so you take a moment and work out the destination. Given that you can only walk on the street grid of the city, how far is the shortest path to the destination?

For example:

> Following R2, L3 leaves you 2 blocks East and 3 blocks North, or 5 blocks away.
>
> R2, R2, R2 leaves you 2 blocks due South of your starting position, which is 2 blocks away.
>
> R5, L5, R5, R3 leaves you 12 blocks away.

How many blocks away is Easter Bunny HQ?

Problem 3 : Validate IP Address

Given a string queryIP, return "IPv4" if IP is a valid IPv4 address, "IPv6" if IP is a valid IPv6 address or "Neither" if IP is not a correct IP of any type.

A valid IPv4 address is an IP in the form "x1.x2.x3.x4" where 0 <= xi <= 255 and xi cannot contain leading zeros. For example, "192.168.1.1" and "192.168.1.0" are valid IPv4 addresses while "192.168.01.1", "192.168.1.00", and "192.168@1.1" are invalid IPv4 addresses.

A valid IPv6 address is an IP in the form "x1:x2:x3:x4:x5:x6:x7:x8" where:

- 1 <= xi.length <= 4
- xi is a hexadecimal string which may contain digits, lowercase English letter ('a' to 'f') and upper-case English letters ('A' to 'F').
- Leading zeros are allowed in xi.

For example, "2001:0db8:85a3:0000:0000:8a2e:0370:7334" and "2001:db8:85a3:0:0:8A2E:0370:7334" are valid IPv6 addresses, while "2001:0db8:85a3::8A2E:037j:7334" and "02001:0db8:85a3:0000:0000:8a2e:0370:7334" are invalid IPv6 addresses.

Example 1:
Input: queryIP = "172.16.254.1"
Output: "IPv4"
Explanation: This is a valid IPv4 address, return "IPv4".

Example 2:
Input: queryIP = "2001:0db8:85a3:0:0:8A2E:0370:7334"
Output: "IPv6"
Explanation: This is a valid IPv6 address, return "IPv6".

Example 3:
Input: queryIP = "256.256.256.256"
Output: "Neither"
Explanation: This is neither a IPv4 address nor a IPv6 address.

Constraints:

- queryIP consists only of English letters, digits and the characters '.' and ':'.

Problem 4 : Next_permutation : find next lexicographically greater permutation

Problem Statement: Given an array Arr[] of integers, rearrange the numbers of the given array into the lexicographically next greater permutation of numbers.
If such an arrangement is not possible, it must rearrange to the lowest possible order (i.e., sorted in ascending order).
Examples
Example 1 :
Input format: Arr[] = {1,3,2}
Output: Arr[] = {2,1,3}

Explanation: All permutations of {1,2,3} are {{1,2,3} , {1,3,2}, {2,13} , {2,3,1} , {3,1,2} , {3,2,1}}. So, the next permutation just after {1,3,2} is {2,1,3}.
Example 2:
Input format: Arr[] = {3,2,1}
Output: Arr[] = {1,2,3}
Explanation: As we see all permutations of {1,2,3}, we find {3,2,1} at the last position. So, we have to return the topmost permutation.

Problem 5 : Next Greater Element I

Easy
TopicsCompanies
The next greater element of some element x in an array is the first greater element that is to the right of x in the same array.
You are given two distinct 0-indexed integer arrays nums1 and nums2, where nums1 is a subset of nums2.
For each 0 <= i < nums1.length, find the index j such that nums1[i] == nums2[j] and determine the next greater element of nums2[j] in nums2. If there is no next greater element, then the answer for this query is -1.
Return an array ans of length nums1.length such that ans[i] is the next greater element as described above.

Example 1:
Input: nums1 = [4,1,2], nums2 = [1,3,4,2]
Output: [-1,3,-1]
Explanation: The next greater element for each value of nums1 is as follows:
- 4 is underlined in nums2 = [1,3,4,2]. There is no next greater element, so the answer is -1.
- 1 is underlined in nums2 = [1,3,4,2]. The next greater element is 3.
- 2 is underlined in nums2 = [1,3,4,2]. There is no next greater element, so the answer is -1.

Example 2:
Input: nums1 = [2,4], nums2 = [1,2,3,4]
Output: [3,-1]
Explanation: The next greater element for each value of nums1 is as follows:
- 2 is underlined in nums2 = [1,2,3,4]. The next greater element is 3.

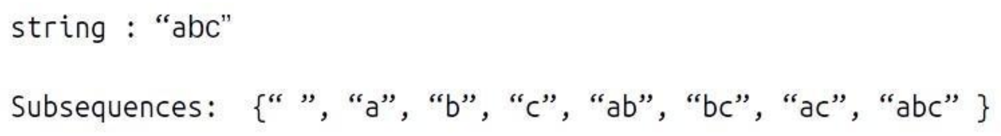
- 4 is underlined in nums2 = [1,2,3,4]. There is no next greater element, so the answer is -1.

Problem 6 : Longest Common Subsequence

Before proceeding further, let us understand what is the “Longest Common Subsequence”, or rather what is a “**subsequence**”?

A subsequence of a string is a list of characters of the string where some characters are deleted (or not deleted at all) and they should be in the same order in the subsequence as in the original string.

For eg:



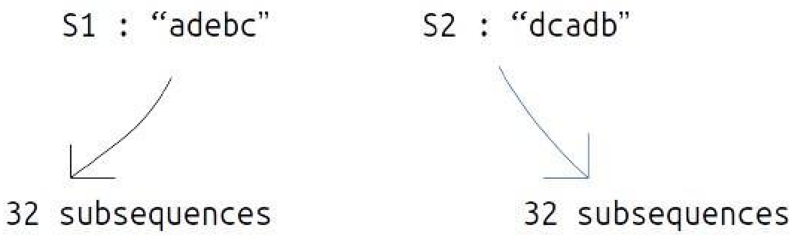
```
string : “abc”

Subsequences:  {“ ”, “a”, “b”, “c”, “ab”, “bc”, “ac”, “abc” }
```

Strings like “cab”,” bc” will not be called as a subsequence of “abc” as the characters are not coming in the same order.

Note: For a string of length n, the number of subsequences will be **2n**.

Now we will look at “Longest Common Subsequence”. The longest Common Subsequence is defined for two strings. It is the common subsequence that has the greatest length.

Examples



S1 : “adebc”

S2 : “dcadb”

32 subsequences

32 subsequences

Longest common subsequence : “adb”