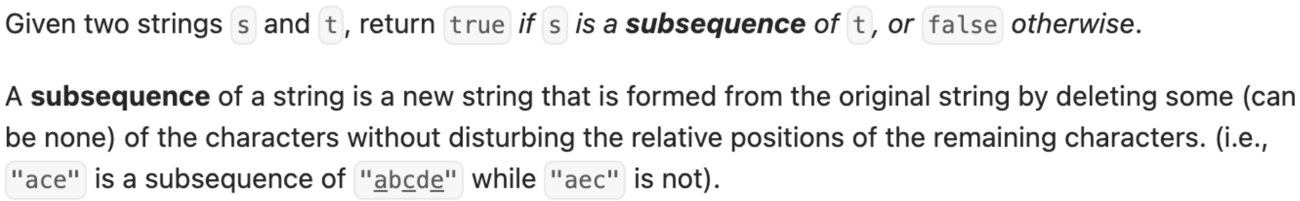
2023-11-11 - Handout – Leetcode Top 100

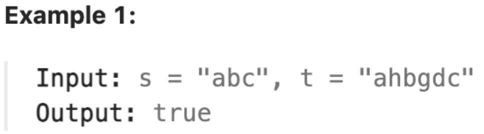
Q1. Is Subsequence

Link: https://leetcode.com/problems/is-subsequence/description/?envType=study-plan-v2&envId=top-interview-150



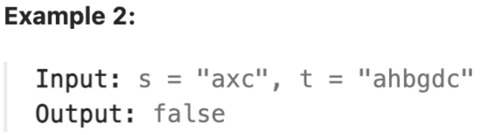
Given two strings `s` and `t`, return `true` *if* `s` *is a* ***subsequence*** *of* `t`*, or* `false` *otherwise*.

A **subsequence** of a string is a new string that is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (i.e., `"ace"` is a subsequence of `"abcde"` while `"aec"` is not).



Example 1:

```
Input: s = "abc", t = "ahbgdc"
Output: true
```

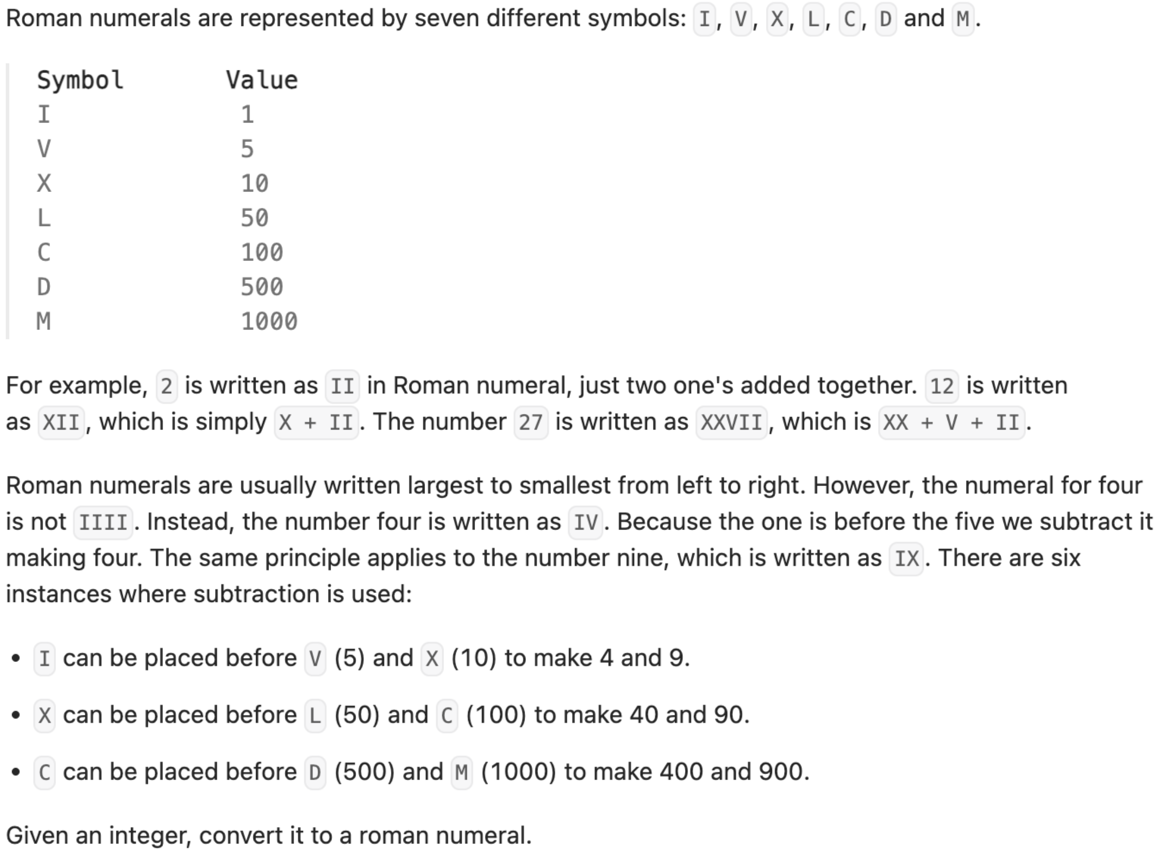


Example 2:

```
Input: s = "axc", t = "ahbgdc"
Output: false
```

Q2. Integer to Roman

Link: https://leetcode.com/problems/integer-to-roman/description/?envType=study-plan-v2&envId=top-interview-150



Roman numerals are represented by seven different symbols: `I`, `V`, `X`, `L`, `C`, `D` and `M`.

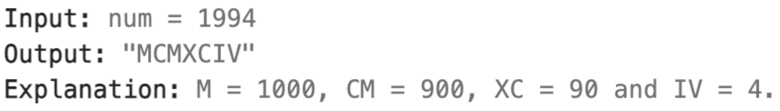
Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, `2` is written as `II` in Roman numeral, just two one's added together. `12` is written as `XII`, which is simply `X + II`. The number `27` is written as `XXVII`, which is `XX + V + II`.

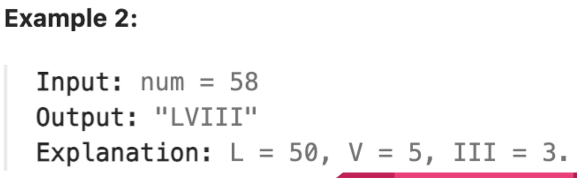
Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not `IIII`. Instead, the number four is written as `IV`. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as `IX`. There are six instances where subtraction is used:

- `I` can be placed before `V` (5) and `X` (10) to make 4 and 9.
- `X` can be placed before `L` (50) and `C` (100) to make 40 and 90.
- `C` can be placed before `D` (500) and `M` (1000) to make 400 and 900.

Given an integer, convert it to a roman numeral.



```
Input: num = 1994
Output: "MCMXCIV"
Explanation: M = 1000, CM = 900, XC = 90 and IV = 4.
```



Example 2:

```
Input: num = 58
Output: "LVIII"
Explanation: L = 50, V = 5, III = 3.
```

Q3. Simplify Path

Link: https://leetcode.com/problems/simplify-path/description/?envType=study-plan-v2&envId=top-interview-150

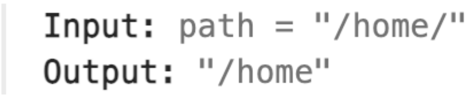
Given a string `path`, which is an **absolute path** (starting with a slash `'/'`) to a file or directory in a Unix-style file system, convert it to the simplified **canonical path**.

In a Unix-style file system, a period `'.'` refers to the current directory, a double period `'..'` refers to the directory up a level, and any multiple consecutive slashes (i.e. `'//'`) are treated as a single slash `'/'`. For this problem, any other format of periods such as `'...'` are treated as file/directory names.

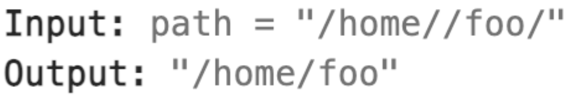
The **canonical path** should have the following format:

- The path starts with a single slash `'/'`.
- Any two directories are separated by a single slash `'/'`.
- The path does not end with a trailing `'/'`.
- The path only contains the directories on the path from the root directory to the target file or directory (i.e., no period `'.'` or double period `'..'`)

Return *the simplified **canonical path***.



```
Input: path = "/home/"
Output: "/home"
```



```
Input: path = "/home//foo/"
Output: "/home/foo"
```

Q4. Product of Array Except Self

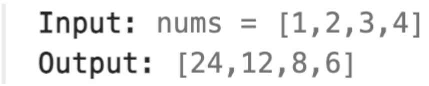
Link: https://leetcode.com/problems/product-of-array-except-self/description/?envType=study-plan-v2&envId=top-interview-150

Given an integer array `nums`, return *an array* `answer` *such that* `answer[i]` *is equal to the product of all the elements of* `nums` *except* `nums[i]`.

The product of any prefix or suffix of `nums` is **guaranteed** to fit in a **32-bit** integer.

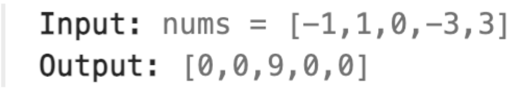
You must write an algorithm that runs in `O(n)` time and without using the division operation.

Example 1:



```
Input: nums = [1,2,3,4]
Output: [24,12,8,6]
```

Example 2:



```
Input: nums = [-1,1,0,-3,3]
Output: [0,0,9,0,0]
```