

### 11. Container With Most Water

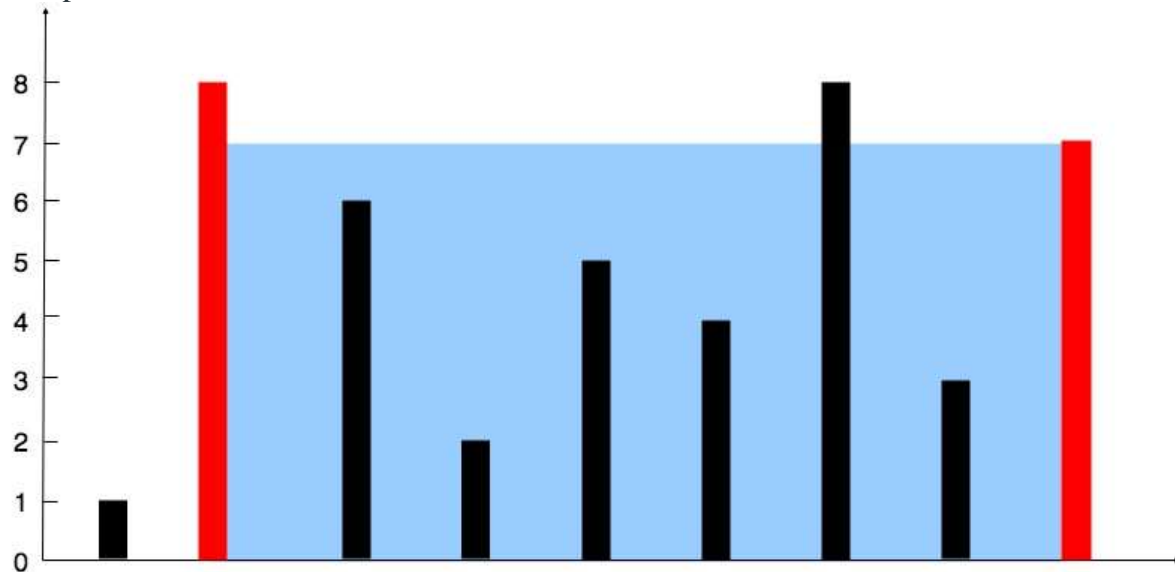
You are given an integer array `height` of length `n`. There are `n` vertical lines drawn such that the two endpoints of the `i`th line are `(i, 0)` and `(i, height[i])`.

Find two lines that together with the x-axis form a container, such that the container contains the most water.

Return *the maximum amount of water a container can store*.

Notice that you may not slant the container.

Example 1:



Input: `height = [1,8,6,2,5,4,8,3,7]`

Output: 49

Explanation: The above vertical lines are represented by array `[1,8,6,2,5,4,8,3,7]`. In this case, the max area of water (blue section) the container can contain is 49.

Example 2:

Input: `height = [1,1]`

Output: 1

Constraints:

- `n == height.length`
- `2 <= n <= 105`
- `0 <= height[i] <= 104`

### 12. Integer to Roman

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.

Example 1:

Input: num = 3

Output: "III"

Explanation: 3 is represented as 3 ones.

Example 2:

Input: num = 58

Output: "LVIII"

Explanation: L = 50, V = 5, III = 3.

Example 3:

Input: num = 1994

Output: "MCMXCIV"

Explanation: M = 1000, CM = 900, XC = 90 and IV = 4.

Constraints:

- $1 \leq \text{num} \leq 3999$

### 13. Roman to Integer

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 ones 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 a roman numeral, convert it to an integer.

Example 1:

Input: s = "III"

Output: 3

Explanation: III = 3.

Example 2:

Input: s = "LVIII"

Output: 58

Explanation: L = 50, V = 5, III = 3.

Example 3:

Input: s = "MCMXCIV"

Output: 1994

Explanation: M = 1000, CM = 900, XC = 90 and IV = 4.

Constraints:

- 1 <= s.length <= 15
- s contains only the characters ('I', 'V', 'X', 'L', 'C', 'D', 'M').
- It is guaranteed that s is a valid roman numeral in the range [1, 3999].

#### 14. Longest Common Prefix

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.

Constraints:

- $1 \leq \text{strs.length} \leq 200$
- $0 \leq \text{strs}[i].\text{length} \leq 200$
- $\text{strs}[i]$  consists of only lowercase English letters.

### 15. 3Sum

Given an integer array `nums`, return all the triplets  $[\text{nums}[i], \text{nums}[j], \text{nums}[k]]$  such that  $i \neq j$ ,  $i \neq k$ , and  $j \neq k$ , and  $\text{nums}[i] + \text{nums}[j] + \text{nums}[k] == 0$ .

Notice that the solution set must not contain duplicate triplets.

Example 1:

Input: `nums = [-1,0,1,2,-1,-4]`

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

Explanation:

$\text{nums}[0] + \text{nums}[1] + \text{nums}[2] = (-1) + 0 + 1 = 0$ .

$\text{nums}[1] + \text{nums}[2] + \text{nums}[4] = 0 + 1 + (-1) = 0$ .

$\text{nums}[0] + \text{nums}[3] + \text{nums}[4] = (-1) + 2 + (-1) = 0$ .

The distinct triplets are `[-1,0,1]` and `[-1,-1,2]`.

Notice that the order of the output and the order of the triplets does not matter.

Example 2:

Input: `nums = [0,1,1]`

Output: `[]`

Explanation: The only possible triplet does not sum up to 0.

Example 3:

Input: `nums = [0,0,0]`

Output: `[[0,0,0]]`

Explanation: The only possible triplet sums up to 0.

Constraints:

- $3 \leq \text{nums.length} \leq 3000$
- $-105 \leq \text{nums}[i] \leq 105$

### 16. 3Sum Closest

Given an integer array `nums` of length `n` and an integer `target`, find three integers in `nums` such that the sum is closest to `target`.

Return *the sum of the three integers*.

You may assume that each input would have exactly one solution.

Example 1:

Input: `nums = [-1,2,1,-4]`, `target = 1`

Output: 2

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

Example 2:

Input: `nums = [0,0,0]`, `target = 1`

Output: 0

Explanation: The sum that is closest to the target is 0. ( $0 + 0 + 0 = 0$ ).

Constraints:

- $3 \leq \text{nums.length} \leq 500$
- $-1000 \leq \text{nums}[i] \leq 1000$
- $-104 \leq \text{target} \leq 104$

### 17. Letter Combinations of a Phone Number

Given a string containing digits from 2-9 inclusive, return all possible letter combinations that the number could represent. Return the answer in any order.

A mapping of digits to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.



Example 1:

Input: digits = "23"

Output: ["ad","ae","af","bd","be","bf","cd","ce","cf"]

Example 2:

Input: digits = ""

Output: []

Example 3:

Input: digits = "2"

Output: ["a","b","c"]

Constraints:

- $0 \leq \text{digits.length} \leq 4$
- $\text{digits}[i]$  is a digit in the range ['2', '9'].

### 18. 4Sum

Given an array `nums` of  $n$  integers, return an array of all the unique quadruplets `[nums[a], nums[b], nums[c], nums[d]]` such that:

- $0 \leq a, b, c, d < n$
- $a, b, c,$  and  $d$  are distinct.
- $\text{nums}[a] + \text{nums}[b] + \text{nums}[c] + \text{nums}[d] == \text{target}$

You may return the answer in any order.

Example 1:

Input:  $\text{nums} = [1,0,-1,0,-2,2]$ ,  $\text{target} = 0$

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

Example 2:

Input:  $\text{nums} = [2,2,2,2,2]$ ,  $\text{target} = 8$

Output:  $[[2,2,2,2]]$

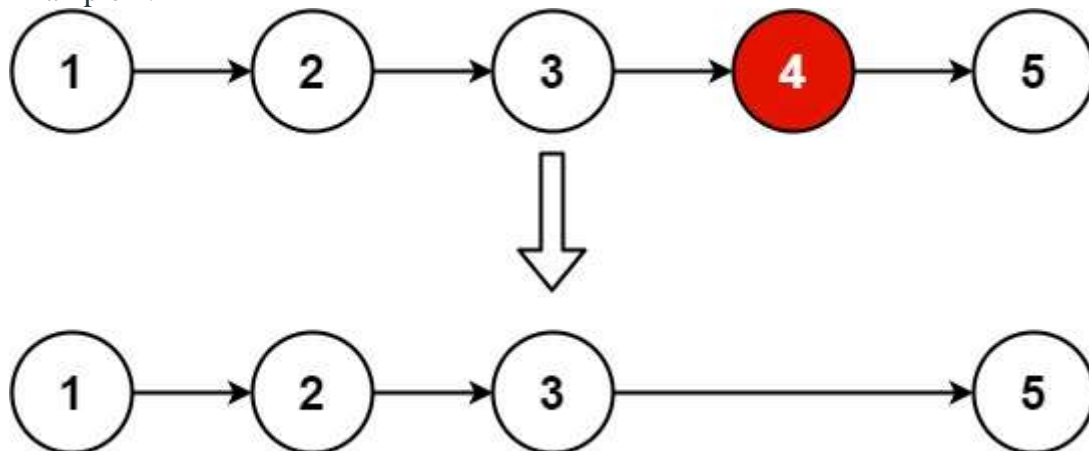
Constraints:

- $1 \leq \text{nums.length} \leq 200$
- $-109 \leq \text{nums}[i] \leq 109$
- $-109 \leq \text{target} \leq 109$

19. Remove Nth Node From End of List

Given the head of a linked list, remove the  $n$ th node from the end of the list and return its head.

Example 1:



Input:  $\text{head} = [1,2,3,4,5]$ ,  $n = 2$

Output:  $[1,2,3,5]$

Example 2:

Input:  $\text{head} = [1]$ ,  $n = 1$

Output:  $[]$

Example 3:

Input:  $\text{head} = [1,2]$ ,  $n = 1$

Output:  $[1]$

Constraints:

- The number of nodes in the list is `sz`.
- $1 \leq sz \leq 30$
- $0 \leq \text{Node.val} \leq 100$
- $1 \leq n \leq sz$

## 20. Valid Parentheses

Given a string `s` containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.
3. Every close bracket has a corresponding open bracket of the same type.

Example 1:

Input: `s = "()"`

Output: `true`

Example 2:

Input: `s = "()[]{}"`

Output: `true`

Example 3:

Input: `s = "(]"`

Output: `false`

Constraints:

- $1 \leq s.length \leq 104$
- `s` consists of parentheses only '()[]{}'.