

## Unit 1: Basics of Python Programming

### 1. Two Sum (1)

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to target*. You may assume that each input would have *exactly* one solution, and you may not use the *same* element twice. You can return the answer in any order.

Example 1:

Input: `nums = [2,7,11,15]`, `target = 9`

Output: `[0,1]`

Explanation: Because `nums[0] + nums[1] == 9`, we return `[0, 1]`.

Example 2:

Input: `nums = [3,2,4]`, `target = 6`

Output: `[1,2]`

Example 3:

Input: `nums = [3,3]`, `target = 6`

Output: `[0,1]`

### 2. Contains Duplicate (217)

Given an integer array `nums`, return `true` if any value appears at least twice in the array, and return `false` if every element is distinct.

Example 1:

Input: `nums = [1,2,3,1]`

Output: `true`

Explanation:

The element 1 occurs at the indices 0 and 3.

Example 2:

Input: `nums = [1,2,3,4]`

Output: `false`

Explanation:

All elements are distinct.

Example 3:

Input: `nums = [1,1,1,3,3,4,3,2,4,2]`

Output: `true`

### 3. Single Number (136)

Given a non-empty array of integers `nums`, every element appears *twice* except for one. Find that single one.

You must implement a solution with a linear runtime complexity and use only constant extra space. (136)

Example 1:

Input: `nums = [2,2,1]`

Output: 1

Example 2:

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

Output: 4

Example 3:

Input: nums = [1]

Output: 1

#### 4. Intersection of Two Arrays (350)

Given two integer arrays nums1 and nums2, return *an array of their intersection*. Each element in the result must appear as many times as it shows in both arrays and you may return the result in any order.

Example 1:

Input: nums1 = [1,2,2,1], nums2 = [2,2]

Output: [2,2]

Example 2:

Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]

Output: [4,9]

Explanation: [9,4] is also accepted.

#### 5. N-ary Tree Preorder Traversal (589)

Given the root of an n-ary tree, return the preorder traversal of its nodes' values.

N-ary-Tree input serialization is represented in their level order traversal. Each group of children is separated by the null value (See examples)

Example 1:

Input: root = [1,null,3,2,4,null,5,6]

Output: [1,3,5,6,2,4]

Example 2:

Input: root =

[1,null,2,3,4,5,null,null,6,7,null,8,null,9,10,null,null,11,null,12,null,13,null,null,14]

Output: [1,2,3,6,7,11,14,4,8,12,5,9,13,10]

#### 6. Pascal's Triangle (118)

Given an integer numRows, return the first numRows of Pascal's triangle.

In Pascal's triangle, each number is the sum of the two numbers directly above it as shown:

Example 1:

Input: numRows = 5

Output: [[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]

Example 2:

Input: numRows = 1

Output: [[1]]

#### 7. Best Time to Buy and Sell Stock (121)

You are given an array prices where prices[i] is the price of a given stock on the i<sup>th</sup> 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 = 0.

### 8.Reverse String (344)

Write a function that reverses a string. The input string is given as an array of characters s. You must do this by modifying the input array [in-place](#) with O(1) extra memory.

Example 1:

Input: s = ["h","e","l","l","o"]

Output: ["o","l","l","e","h"]

Example 2:

Input: s = ["H","a","n","n","a","h"]

Output: ["h","a","n","n","a","H"]

### 9.Word Pattern (290)

Given a pattern and a string s, find if s follows the same pattern.

Here follow means a full match, such that there is a bijection between a letter in pattern and a non-empty word in s. Specifically:

- Each letter in pattern maps to exactly one unique word in s.
- Each unique word in s maps to exactly one letter in pattern.
- No two letters map to the same word, and no two words map to the same letter.

Example 1:

Input: pattern = "abba", s = "dog cat cat dog"

Output: true

Explanation:

The bijection can be established as:

- 'a' maps to "dog".
- 'b' maps to "cat".

Example 2:

Input: pattern = "abba", s = "dog cat cat fish"

Output: false

Example 3:

Input: pattern = "aaaa", s = "dog cat cat dog"

Output: false

## 10. Design HashSet (705)

Design a HashSet without using any built-in hash table libraries.

Implement MyHashSet class:

- void add(key) Inserts the value key into the HashSet.
- bool contains(key) Returns whether the value key exists in the HashSet or not.
- void remove(key) Removes the value key in the HashSet. If key does not exist in the HashSet, do nothing.

Example 1:

Input

["MyHashSet", "add", "add", "contains", "contains", "add", "contains", "remove", "contains"]  
[[], [1], [2], [1], [3], [2], [2], [2], [2]]

Output

[null, null, null, true, false, null, true, null, false]

Explanation

```
MyHashSet myHashSet = new MyHashSet();  
myHashSet.add(1);    // set = [1]  
myHashSet.add(2);    // set = [1, 2]  
myHashSet.contains(1); // return True  
myHashSet.contains(3); // return False, (not found)  
myHashSet.add(2);    // set = [1, 2]  
myHashSet.contains(2); // return True  
myHashSet.remove(2);  // set = [1]  
myHashSet.contains(2); // return False, (already removed)
```

## 11.Valid Anagram (242)

Given two strings s and t, return true if t is an anagram of s, and false otherwise.

Example 1:

Input: s = "anagram", t = "nagaram"

Output: true

Example 2:

Input: s = "rat", t = "car"

Output: false

## 12.N-ary Tree Preorder Traversal (589)

Given the root of an n-ary tree, return *the preorder traversal of its nodes' values*.

N-ary-Tree input serialization is represented in their level order traversal. Each group of children is separated by the null value (See examples)

Example 1:

Input: root = [1,null,3,2,4,null,5,6]

Output: [1,3,5,6,2,4]

Example 2:

Input: root =

[1,null,2,3,4,5,null,null,6,7,null,8,null,9,10,null,null,11,null,12,null,13,null,null,14]

Output: [1,2,3,6,7,11,14,4,8,12,5,9,13,10]

### 13. Construct Binary Tree from Preorder and Inorder Traversal (105)

Given two integer arrays preorder and inorder where preorder is the preorder traversal of a binary tree and inorder is the inorder traversal of the same tree, construct and return *the binary tree*.

Example 1:

Input: preorder = [3,9,20,15,7], inorder = [9,3,15,20,7]

Output: [3,9,20,null,null,15,7]

Example 2:

Input: preorder = [-1], inorder = [-1]

Output: [-1]

### 14. 3Sum (15)

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

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:

$nums[0] + nums[1] + nums[2] = (-1) + 0 + 1 = 0$ .

$nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0$ .

$nums[0] + nums[3] + nums[4] = (-1) + 2 + (-1) = 0$ .

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.

### 15. Plus One (66)

You are given a large integer represented as an integer array digits, where each digits[i] is the  $i^{\text{th}}$  digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading 0's.

Increment the large integer by one and return *the resulting array of digits*.

Example 1:

Input: digits = [1,2,3]

Output: [1,2,4]

Explanation: The array represents the integer 123.

Incrementing by one gives  $123 + 1 = 124$ .

Thus, the result should be [1,2,4].

Example 2:

Input: digits = [4,3,2,1]

Output: [4,3,2,2]

Explanation: The array represents the integer 4321.

Incrementing by one gives  $4321 + 1 = 4322$ .

Thus, the result should be [4,3,2,2].

Example 3:

Input: digits = [9]

Output: [1,0]

Explanation: The array represents the integer 9.

Incrementing by one gives  $9 + 1 = 10$ .

Thus, the result should be [1,0].

### **16.Pascal's Triangle (118)**

Given an integer numRows, return the first numRows of Pascal's triangle.

In Pascal's triangle, each number is the sum of the two numbers directly above it as shown:

Example 1:

Input: numRows = 5

Output: [[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]

Example 2:

Input: numRows = 1

Output: [[1]]

### **17.Longest Substring Without Repeating Characters (3)**

Given a string s, find the length of the longest substring without duplicate characters.

Example 1:

Input: s = "abcabcbb"

Output: 3

Explanation: The answer is "abc", with the length of 3.

Example 2:

Input: s = "bbbbbb"

Output: 1

Explanation: The answer is "b", with the length of 1.

Example 3:

Input: s = "pwwkew"

Output: 3

Explanation: The answer is "wke", with the length of 3.

Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.

### **18.First Unique Character in a String (387)**

Given a string s, find the first non-repeating character in it and return its index. If it does not exist, return -1.

Example 1:

Input: s = "leetcode"

Output: 0

Explanation:

The character 'l' at index 0 is the first character that does not occur at any other index.

Example 2:

Input: s = "loveleetcode"

Output: 2

Example 3:

Input: s = "aabb"

Output: -1

### 19.Group Anagrams (49)

Given a string s, find the first non-repeating character in it and return its index. If it does not exist, return -1.

Example 1:

Input: s = "leetcode"

Output: 0

Explanation:

The character 'l' at index 0 is the first character that does not occur at any other index.

Example 2:

Input: s = "loveleetcode"

Output: 2

Example 3:

Input: s = "aabb"

Output: -1

### 20.Intersection of Two Arrays (350)

Given two integer arrays nums1 and nums2, return *an array of their intersection*. Each element in the result must appear as many times as it shows in both arrays and you may return the result in any order.

Example 1:

Input: nums1 = [1,2,2,1], nums2 = [2,2]

Output: [2,2]

Example 2:

Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]

Output: [4,9]

Explanation: [9,4] is also accepted.

