cybercom creation

JavaScript Practice 1.5

Subject: Arrays, Strings
Created By: Pritey Mehta

Case 1:

A **valid IP address** consists of exactly four integers separated by single dots. Each integer is between 0 and 255 (**inclusive**) and cannot have leading zeros.

For example, "0.1.2.201" and "192.168.1.1" are **valid** IP addresses, but "0.011.255.245", "192.168.1.312" and "192.168@1.1" are **invalid** IP addresses.

Given a string s containing only digits, return all possible valid IP addresses that can be formed by inserting dots into s. You are **not** allowed to reorder or remove any digits in s. You may return the valid IP addresses in **any** order.

Example 1:

```
Input: s = "25525511135"
Output: ["255.255.11.135","255.255.111.35"]
```

Example 2:

```
Input: s = "0000"
Output: ["0.0.0.0"]
```

Example 3:

```
Input: s = "101023"
Output:
["1.0.10.23","1.0.102.3","10.1.0.23","10.10.2.3","101.0.2.3"]
```

Case 2:

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 = [3,2,-4,-1], target = -4
Output: -3
Explanation: The sum that is closest to the target is 2. (2 + -4 + -1 = -3).

Case 3:

Given an unsorted integer array nums, return the smallest missing positive integer.

You must implement an algorithm that runs in O(n) time and uses constant extra space.

Example 1:

Input: nums = [1,2,0]
Output: 3
Explanation: The numbers in the range [1,2] are all in the array.

Example 2:

Input: nums = [3,4,-1,1]
Output: 2
Explanation: 1 is in the array but 2 is missing.

Example 3:

Input: nums = [7,8,9,11,12]
Output: 1
Explanation: The smallest positive integer 1 is missing.