Problem 93: Restore IP Addresses

Problem Information

Difficulty: Medium
Acceptance Rate: 54.56%
Paid Only: No
Tags: String, Backtracking

Problem Description

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"]

Constraints:

* `1 <= s.length <= 20` * `s` consists of digits only.

Code Snippets

C++:

```
class Solution {
public:
vector<string> restoreIpAddresses(string s) {

}
};
```

Java:

```
class Solution {
public List<String> restoreIpAddresses(String s) {

}
}
```

Python3:

```
class Solution:
def restoreIpAddresses(self, s: str) -> List[str]:
```