Documentation for the restoreIpAddresses Function

Overview

The restoreIpAddresses function takes a string of digits and returns all possible valid IP addresses that can be formed by inserting dots into the string. An IP address is considered valid if it consists of exactly four integers separated by single dots, where each integer is between 0 and 255 and cannot have leading zeros.

Function Signature

def restoreIpAddresses(self, s: str) > List[str]:

Parameters

- **s (str):** A string consisting only of digits.

Returns

- **List[str]:** A list of strings, each representing a valid IP address.

Helper Function: is_valid

This function checks whether a given segment of the IP address is valid:

- The segment should be between 0 and 255.
- The segment should not have leading zeros unless it is exactly 0.

```
def is_valid(segment: str) > bool:
    return 0 <= int(segment) <= 255 and (segment == "0" or not segment.startswith("0"))
```

Helper Function: backtrack

- *This function uses backtracking to explore all possible ways to insert dots into the string to form valid IP addresses:*
  - It starts at a given index in the string and constructs segments of 1 to 3 characters.
  - If a segment is valid, it continues to the next part of the string.
  - Once four segments are collected and the end of the string is reached, a valid IP address is formed and added to the result list.

Main Function: restoreIpAddresses

This function initializes the result list and starts the backtracking process from the beginning of the string.

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.