

# 093. Restore IP Addresses

## 093 Restore IP Addresses

- Backtracking+string

### Description

Given a string containing only digits, restore it by returning all possible valid IP address combinations.

For example:

Given "25525511135" ,

return ["255.255.11.135", "255.255.111.35"] . (Order does not matter)

### 1. Thought line

1. IP Address Format
- 2.

### 2. Backtracking+string

```
1 class Solution {
2 public:
3     vector<string> restoreIpAddresses(string s) {
4         vector<string> ipComb;
5         vector<string> ipNum;
6         findIPAddr(s, 0, ipNum, ipComb);
7         return ipComb;
8     }
9
10    void findIPAddr(string &s, int index, vector<string> &ipNum, vector<string> &ipComb) {
11        if(ipNum.size()==4) {
12            if(index==s.size()) {
13                string ipAddr = ipNum[0];
14                for(int i=1; i<4; i++)
15                    ipAddr += ("."+ipNum[i]);
16                ipComb.push_back(ipAddr);
17            }
18            return;
19        }
20
21        string curNum;
22        for(int i=index; i<s.size() && i<index+3; i++) {
23            curNum.push_back(s[i]);
24            if(isValidNum(curNum)) {
25                ipNum.push_back(curNum);
26                findIPAddr(s, i+1, ipNum, ipComb);
27                ipNum.pop_back();
28            }
29        }
30    }
31
32    bool isValidNum(string s) {
33        if(s.empty() || s.size(>3) return false;
34        if(s[0]=='0' && s.size()!=1) return false;
35        if(s.size()==3 && stoi(s)>255) return false;
36        return true;
37    }
38 };
```

