

```

1  /*
2  给定一个只包含数字的字符串，复原它并返回所有可能的 IP 地址格式。
3
4  示例：
5
6  输入："25525511135"
7  输出：["255.255.11.135", "255.255.111.35"]
8
9  来源：力扣（LeetCode）
10 链接：https://leetcode-cn.com/problems/restore-ip-addresses
11 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
12 */

```

分析:

- 一个byte一个byte的填,当取数据长度为3时要注意对应的整数是否大于了255;
- 若取的数据长度大于1,要去掉第一个数据是否0;
- 递归写法.

方法一:C++\_递归写法

```

1  class Solution
2  {
3      private:
4          void helper(    vector<string>&    vs            ,
5                          string&            s            ,
6                          string            cur_str        ,
7                          int                str_idx        ,
8                          int                count
9                      )
10     {
11         if(count == 4 && str_idx==s.size())
12         {
13             vs.push_back(cur_str);
14             return;
15         }
16
17         for(int length = 1 ; length <=3 ; length++ )
18         {
19             if(s.size() <str_idx+length)
20             {
21                 break;
22             }
23             string sub = s.substr(str_idx,length);
24             /*去掉前导0的可能性*/
25             if(length > 1 && sub[0] == '0')
26             {

```

```

27         continue;
28     }
29
30     if(stoi(sub) > 255)
31     {
32         break;
33     }
34     if(count==3)
35     {
36         helper(vs,s,cur_str+sub,str_idx+length,count+1);
37     }
38     else
39     {
40         helper(vs,s,cur_str+sub+".",str_idx+length,count+1);
41     }
42
43     }
44
45
46
47     }
48     public:
49     vector<string> restoreIpAddresses(string s)
50     {
51         vector<string> vs;
52         if(s.size() < 4 || s.size() > 12)
53         {
54             return vs;
55         }
56
57         helper(vs,s,string(),0,0);
58
59         return vs;
60
61     }
62 }
63 };
64
65 /*
66 执行结果:
67 通过
68 显示详情
69 执行用时 :8 ms, 在所有 cpp 提交中击败了46.22% 的用户
70 内存消耗 :9.3 MB, 在所有 cpp 提交中击败了10.32%的用户
71 */

```