

```

1  /*
2  给定一个仅包含数字 2-9 的字符串，返回所有它能表示的字母组合。
3
4  给出数字到字母的映射如下（与电话按键相同）。注意 1 不对应任何字母。
5
6  示例：
7
8  输入："23"
9  输出：["ad", "ae", "af", "bd", "be", "bf", "cd", "ce", "cf"].
10
11 说明：
12 尽管上面的答案是按字典序排列的，但是你可以任意选择答案输出的顺序。
13
14 来源：力扣（LeetCode）
15 链接：https://leetcode-cn.com/problems/letter-combinations-of-a-phone-number
16 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。
17 */

```

分析:

- 方法一:回溯穷举法递归版本

方法一:C++\_回溯穷举递归版本

```

1  class solution
2  {
3      private:
4          map<char,string> mcs;
5
6
7          void helper(    vector<string>&    vs    ,
8                        string&            digits ,
9                        string&            cur    ,
10                       int                idx
11                    )
12        {
13            if(idx == digits.size())
14            {
15                vs.push_back(cur);
16                return;
17            }
18
19
20            for(int i = 0 ; i < mcs[digits[idx]].size() ; i++)
21            {
22                cur[idx] = mcs[digits[idx]][i];
23                helper(vs,digits,cur,idx+1);
24            }
25        }
26
27
28     public:

```

```

29
30     solution()
31     {
32         mcs['0'] = string("");
33         mcs['1'] = string("*");
34         mcs['2'] = string("abc");
35         mcs['3'] = string("def");
36         mcs['4'] = string("ghi");
37         mcs['5'] = string("jkl");
38         mcs['6'] = string("mno");
39         mcs['7'] = string("pqrs");
40         mcs['8'] = string("tuv");
41         mcs['9'] = string("wxyz");
42     }
43
44     vector<string> letterCombinations(string digits)
45     {
46         vector<string> vs;
47
48         if(digits.size() < 1)
49         {
50             return vs;
51         }
52
53         string cur(digits.size(), ' ');
54         helper(vs, digits, cur, 0);
55         return vs;
56     }
57 };
58
59 /*
60 执行结果:
61 通过
62 显示详情
63 执行用时 :4 ms, 在所有 cpp 提交中击败了76.09% 的用户
64 内存消耗 :8.8 MB, 在所有 cpp 提交中击败了23.27%的用户
65 */

```