

```

1  /*
2  给出 n 代表生成括号的对数，请你写出一个函数，使其能够生成所有可能的并且有效的括号组合。
3
4  例如，给出 n = 3，生成结果为：
5
6  [
7    "((()))",
8    "(()())",
9    "(()())",
10   "()(())",
11   "()()()"
12  ]
13
14  来源：力扣（LeetCode）
15  链接：https://leetcode-cn.com/problems/generate-parentheses
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+剪枝

方法一:C++_回溯+剪枝

```

1  class solution
2  {
3      private:
4
5          void helper (    vector<string>&    vs            ,
6                          string&            cur_s        ,
7                          int                left_num       ,
8                          int                right_num      ,
9                          int                idx            ,
10                         int                n
11                     )
12     {
13
14         if(right_num > left_num)    /*剪枝*/
15         {
16             return;
17         }
18         if( left_num == n && right_num == n)
19         {
20             vs.push_back(cur_s);
21             return;
22         }
23
24         if(idx == 2*n)
25         {
26             return ;
27         }
28

```

```

29
30         cur_s[idx] = '(';
31         helper(vs, cur_s, left_num+1, right_num, idx+1, n);
32         cur_s[idx] = ')';
33         helper(vs, cur_s, left_num, right_num+1, idx+1, n);
34
35     }
36
37
38
39     public:
40     vector<string> generateParenthesis(int n)
41     {
42         vector<string> vs;
43
44         if(n<1)
45         {
46             return vs;
47         }
48
49         string cur_s(2*n, ' ');
50
51
52         helper(vs, cur_s, 0, 0, 0, n);
53
54         return vs;
55
56     }
57 };
58
59
60
61 /*
62 执行结果:
63 通过
64 显示详情
65 执行用时 :8 ms, 在所有 cpp 提交中击败了81.67% 的用户
66 内存消耗 :13.6 MB, 在所有 cpp 提交中击败了91.98%的用户
67 */
68

```