# 题目

数字n代表生成括号的对数，请你设计一个函数，用于能够生成所有可能的并且有效的括号组合。

示例 1：

输入：n = 3

输出：["((()))","(()())","(())()","()(())","()()()"]

示例 2：

输入：n = 1

输出：["()"]

提示：

1 <= n <= 8

# 分析

## 方法一：暴力法

## 方法二：回溯法

分析：

我们可以通过跟踪到目前为止放置的左括号和右括号的数目来做到这一点。

如果左括号数量不大于n，我们可以放一个左括号。如果右括号数量小于左括号的数量，我们可以放一个右括号。

代码：

class Solution {

public:

vector<string> generateParenthesis(int n) {

vector<string> result;

string current;

generate(result, current, 0, 0, n);

return result;

}

private:

void generate(vector<string>& result, string current, int open, int close, int n) {

// 如果生成的字符串长度达到 2\*n，且括号匹配，将其添加到结果中

if (current.length() == 2 \* n) {

result.push\_back(current);

return;

}

// 如果左括号的数量小于 n，可以添加一个左括号

if (open < n) {

current.push\_back('(');

generate(result, current, open + 1, close, n);

current.pop\_back();

}

// 如果右括号的数量小于左括号的数量，可以添加一个右括号

if (close < open) {

current.push\_back(')');

generate(result, current , open, close + 1, n);

current.pop\_back();

}

}

};