22. Generate Parentheses

Medium

Given *n* pairs of parentheses, write a function to generate all combinations of well-formed parentheses.

For example, given *n* = 3, a solution set is:

[  
 "((()))",  
 "(()())",  
 "(())()",  
 "()(())",  
 "()()()"  
]  
class Solution {

public:

vector<string> strings;

int count;

vector<string> generateParenthesis(int n) {

count=n;

helper("",0,0);

return strings;

}

void helper(string current,int left,int right){

if(current.size()==count\*2){

strings.push\_back(current);

return;

}

if(left<count){

helper(current+"(",left+1,right);

}

if(right<left){

helper(current+")",left,right+1);

}

}

};

Success

[Details](https://leetcode.com/submissions/detail/205181964/)

Runtime: 0 ms, faster than 100.00% of C++ online submissions for Generate Parentheses.

Memory Usage: 966.7 KB, less than 33.15% of C++ online submissions for Generate Parentheses.