Intuition

This problem is one of the classical recursion problems.
For any given n, lets say n = 2, we have to fill four places in our output ("_ _ _ _"). And each of these places can be either filled by an open braces "(" or a closed braces ")".

Approach

"_ _ _ _"

/ \

'(' ')'

For every place we have two choices and 1 decision to make.
Our choices are to either use '(' or ')'.

Now lets try to visualize the recursive tree based upon the choices discussed above.

Initially, we have:
For n = 3
current ouput = ""
availableOpenBracketsCnt = 3 and availableCloseBracketsCnt = 3

The first choise is very simple. Since we can not start a balanced parenthesis sequence with ')', we have only one choice in the begining. So our output will be '(' and count of open brackets left = 2 and count of closed brackets left = 3.

op ip

"" O-3, C-3

"(",O-2,C-3

"((",O-1,C-3 "()", O-2,C-2

"(((",0,3 "(()",1,2 "()(",1,2

"((()",0,2 "(()(",0,2 "(())",1,1 "()((",0,2 "()()",1,1

"(((())",0,1 "(()()",0,1 "(())(",0,1 "()(()",0,1 "()()(",0,1

"((()))",0,0 "(()())",0,0 "(())()",0,0 "()(())",0,0 "()()()", 0,0

Observation from the recursive tree

- Whenever we have count of open brackets equal to the count of close brackets, we have only one choice - that is to use '('. Because, all the brackets till now have been balanced. And we can not start a new sequence with ')'.
- Whenever, count of close bracket is 0, we can only use '('.
- Whenever, count of open bracket is 0, we can only use ')'.
- And for all the remaining cases, we have both the choices.
- We get an answer, when count of open == 0 and count of close == 0.