

```

class Solution {
public:
    void extend(int countOpen, int countClose, int n, string s, vector<string>&
AllPairs){
        if (s.length() == 2 * n){
            if (countOpen == countClose)
                AllPairs.push_back(s);

            return;
        }

        for (int i = 0; i <= 1; i++){
            if (i == 0){
                if (countOpen < n)
                    extend(countOpen + 1, countClose, n, s + '(', AllPairs);
            }
            else{
                if (countClose < countOpen)
                    extend(countOpen, countClose + 1, n, s + ')', AllPairs);
            }
        }
    }

    vector<string> generateParenthesis(int n) {
        vector<string> validPairs;
        extend(0,0,n, "", validPairs);
        return validPairs;
    }
};

```