JavaScript Score of Parentheses

Challenge

Given a balanced parentheses string `s`, return the score of the string.

The score of a balanced parentheses string is based on the following rules:

- `'()'` has score `1`.
- `AB` has score `A + B`, where `A` and `B` are balanced parentheses strings.
- `(A)` has score `2 * A`, where `A` is a balanced parentheses string.

1st Example

```
Input: s = '()'
Output: 1
```

2nd Example

```
Input: s = '(())'
Output: 2
```

3rd Example

```
Input: s = '()()'
Output: 2
```

Constraints

- `2 <= s.length <= 50`
- `s` consists of only `'('` and `')'`.
- `s` is a balanced parentheses string.

Solution

```
const scoreOfParentheses = (s) => {
    let score = 0,
        depth = 0;

    for (let i = 0, j = s.length; i < j; i++) {
        if (s.charAt(i) == '(')
            depth++;
        else if (s.charAt(i - 1) == '(')
            score += 1 << --depth;
        else
            --depth;
    }

    return score;
};
```

Explanation

I've built a function called `scoreOfParentheses` that takes a string `s` as input. The purpose of this function is to calculate the score of a valid parentheses sequence represented by the string `s`. The score is based on the depth of the parentheses.

Inside the function, two variables `score` and `depth` are initialized to `0`.

A `for` loop is used to iterate through each character of the string `s`. Within the loop, various conditions are checked to determine the score calculation.

If the current character is an opening parenthesis `'('`, the `depth` variable is incremented by `1` to indicate an increase in the depth of the parentheses.

If the current character is a closing parenthesis `')'` and the previous character is an opening parenthesis `'('`, it means a valid pair of parentheses is found. In this case, the score is calculated by adding `1` shifted left by `depth-1` to the `score` variable. The `depth` variable is then decremented by `1` to indicate a decrease in the depth.

If none of the above conditions are met, it means the current character is a closing parenthesis and the previous character is also a closing parenthesis. In this case, the `depth` variable is decremented by `1` to indicate a decrease in the depth.

After the loop ends, the final calculated `score` is returned as the output of the function.

In summary, this function calculates the score of a valid parentheses sequence by keeping track of the depth of the parentheses. The score is calculated based on the depth, with higher depths contributing more to the score.

Author: Trevor Morin