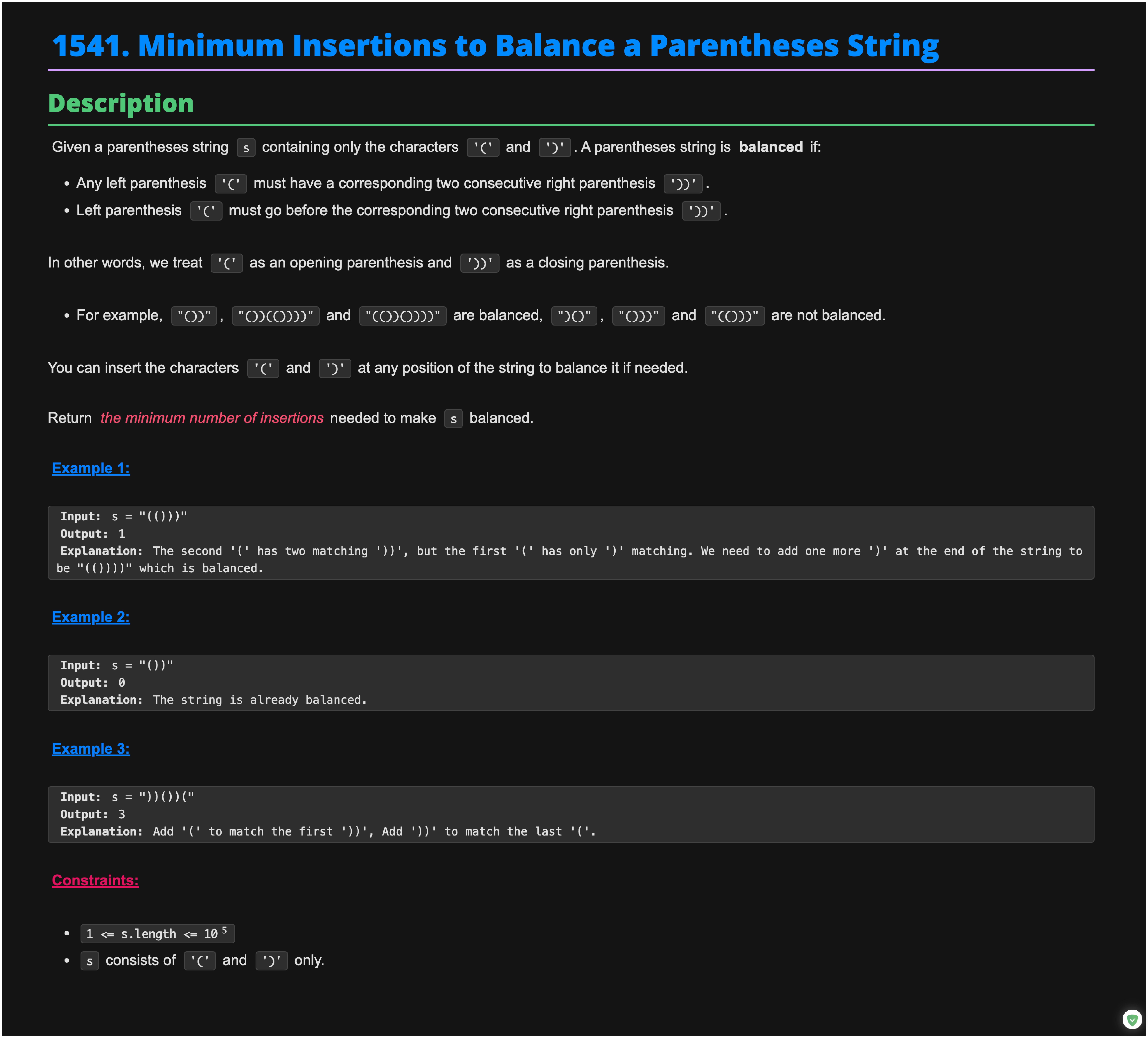

1541. Minimum Insertions to Balance a Parentheses String

Description

Given a parentheses string `s` containing only the characters `'('` and `')'`. A parentheses string is **balanced** if:

- Any left parenthesis `'('` must have a corresponding two consecutive right parenthesis `'))'`.
- Left parenthesis `'('` must go before the corresponding two consecutive right parenthesis `'))'`.

In other words, we treat `'('` as an opening parenthesis and `'))'` as a closing parenthesis.

- For example, `"())"`, `"())(())))"` and `"(())())))"` are balanced, `")()"`, `"()))"` and `"(()))"` are not balanced.

You can insert the characters `'('` and `')'` at any position of the string to balance it if needed.

Return *the minimum number of insertions* needed to make `s` balanced.

Example 1:

```
Input: s = "(()))"
Output: 1
Explanation: The second '(' has two matching '))', but the first '(' has only ')' matching. We need to add one more ')' at the end of the string to
be "(())))" which is balanced.
```

Example 2:

```
Input: s = "())"
Output: 0
Explanation: The string is already balanced.
```

Example 3:

```
Input: s = "))())("
Output: 3
Explanation: Add '(' to match the first '))', Add '))' to match the last '('.
```

Constraints:

- `1 <= s.length <= 10`^{5}
- `s` consists of `'('` and `')'` only.