

# Problem 1190: Reverse Substrings Between Each Pair of Parentheses

## Problem Information

Difficulty: Medium

Acceptance Rate: 71.86%

Paid Only: No

Tags: String, Stack

## Problem Description

You are given a string `s`` that consists of lower case English letters and brackets.

Reverse the strings in each pair of matching parentheses, starting from the innermost one.

Your result should **not** contain any brackets.

**Example 1.**

**Input:** `s = "(abcd)"` **Output:** `"dcba"`

**Example 2.**

**Input:** `s = "(u(love)i)"` **Output:** `"iloveu"` **Explanation:** The substring "love" is reversed first, then the whole string is reversed.

**Example 3.**

**Input:** `s = "(ed(et(oc))el)"` **Output:** `"leetcode"` **Explanation:** First, we reverse the substring "oc", then "etco", and finally, the whole string.

**Constraints:**

`1 <= s.length <= 2000`` `s`` only contains lower case English characters and parentheses.   
It is guaranteed that all parentheses are balanced.

## Code Snippets

### C++:

```
class Solution {  
public:  
    string reverseParentheses(string s) {  
  
    }  
};
```

### Java:

```
class Solution {  
    public String reverseParentheses(String s) {  
  
    }  
}
```

### Python3:

```
class Solution:  
    def reverseParentheses(self, s: str) -> str:
```