

Problem 301: Remove Invalid Parentheses

Problem Information

Difficulty: Hard

Acceptance Rate: 49.58%

Paid Only: No

Tags: String, Backtracking, Breadth-First Search

Problem Description

Given a string `s` that contains parentheses and letters, remove the minimum number of invalid parentheses to make the input string valid.

Return `_` a list of **unique strings** that are valid with the minimum number of removals`_`. You may return the answer in **any order**.

Example 1:

Input: `s = "()())()` **Output:** `["()()()", "()()()"]`

Example 2:

Input: `s = "(a())()"` **Output:** `["(a())()", "(a())()"]`

Example 3:

Input: `s = ")("` **Output:** `[""]`

Constraints:

- `1 <= s.length <= 25` * `s` consists of lowercase English letters and parentheses `'('` and `')'`.
- There will be at most `20` parentheses in `s`.

Code Snippets

C++:

```
class Solution {  
public:  
    vector<string> removeInvalidParentheses(string s) {  
  
    }  
};
```

Java:

```
class Solution {  
    public List<String> removeInvalidParentheses(String s) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def removeInvalidParentheses(self, s: str) -> List[str]:
```