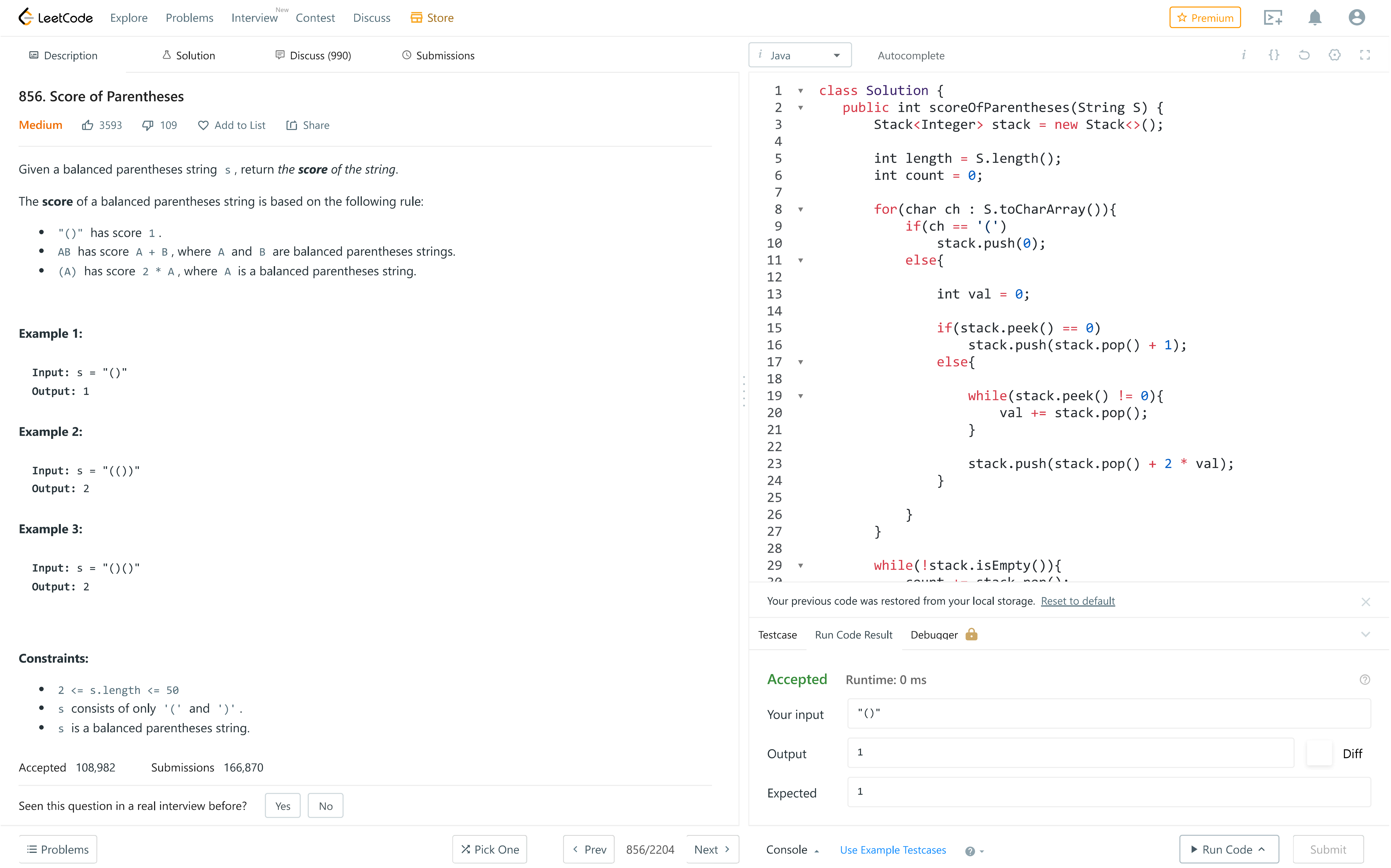

856. Score of Parentheses

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

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

- `"()"` 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.

Example 1:

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

Example 2:

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

Example 3:

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

Constraints:

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

Accepted 108,982 Submissions 166,870

Seen this question in a real interview before? Yes No

```java
class Solution {
    public int scoreOfParentheses(String S) {
        Stack<Integer> stack = new Stack<>();
        
        int length = S.length();
        int count = 0;
        
        for(char ch : S.toCharArray()){
            if(ch == '(')
                stack.push(0);
            else{
                
                int val = 0;
                
                if(stack.peek() == 0)
                    stack.push(stack.pop() + 1);
                else{
                    
                    while(stack.peek() != 0){
                        val += stack.pop();
                    }
                    
                    stack.push(stack.pop() + 2 * val);
                }
                
            }
        }
        
        while(!stack.isEmpty()){
            count += stack.pop();
```

Your previous code was restored from your local storage. Reset to default

Accepted Runtime: 0 ms

Your input	"()"
Output	1
Expected	1