Rotate String

solved by	Senan
Platform	LeetCode
difficulty	Easy
# Serial	796
tags	String Manipulation
language	C++
solved on	@03/11/2024
link	https://leetcode.com/problems/rotate-string/description/
Completion	☑

Intuition

The problem requires checking if one string (`goal`) is a rotation of another string (`s`). The key insight is that if `goal` is indeed a rotation of `s`, then it should appear as a substring within the concatenation of `s` with itself (`s + s`). This approach leverages the properties of string rotation effectively.

Approach

1. Check if the lengths of `s` and `goal` are equal. If not, return `false` immediately because strings of different lengths cannot be rotations of each other.
2. Concatenate `s` with itself to form a new string (`s + s`).
3. Check if `goal` appears as a substring in this concatenated string using the `find()` function.
4. If `goal` is found as a substring, return `true`; otherwise, return `false`.

Complexity

Time Complexity:

- **O(n)**: Where `n` is the length of the string `s`. Concatenation takes `O(n)`, and the `find()` function takes `O(n)` for a substring search.

Space Complexity:

- **O(n)**: The space complexity is `O(n)` due to the concatenated string `s + s`, which is double the size of `s`.

Code

```
class Solution {
public:
    bool rotateString(string s, string goal) {
        return (s.size() == goal.size()) && (s + s).find(goal) != -1;
    }
};
```