



Experiment-1.2

Student Name: Aayush Gurung

Branch: CSE

Semester: 6

Subject Name: CC LAB

UID:20BCS5323

Section/Group:DM_607(A)

Date of Performance:24-02-2023

Subject Code: 20CST-355

Aim:

To demonstrate the concept of String Matching algorithms

Objective

Problem 1:

Rotate-String

Approach:

Checking if the concatenation of s with itself contains goal

Algorithm:

1. Declare a and $a=s+s$;
2. First check if the size of s and goal is same, if not then return false otherwise move to next step
3. Then check if a contains goal or not if not then return false else return true.

Code:

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



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Output:

LeetCode

Problem List

Premium

Close

Accepted

Next question

- 797. All Paths From Source to Target

More challenges

- 1898. Maximum Number of Removable Characters
- 1698. Number of Distinct Substrings in a String
- 616. Add Bold Tag in String

All statuses

All languages

Accepted
a minute ago

C++

Related Tags

Select tags 0/5

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

Console

Run

Submit

Problem 2:

Repeated String Match

Approach:

Add the string into a separate variable until it equals given string if its not equal then return -1;

Algorithm:

1. Declare the int variables to store the size of a and b string
2. Divide both the int variables and add 2 to it to get it equal to the no of times we need to run the loop
3. Then run a loop and constantly concatenate the string a to temp variable and check if b is present or not.
4. If b is present then return count, if not present then return -1;

Code:

```
class Solution {
public:
    int repeatedStringMatch(string a, string b) {
        int la=a.size();
        int lb=b.size();
        int k=(ceil(lb/la))+2;
        int count=0;
        string temp="";
        while(k--){
            temp+=a;
            count++;
            // cout<<temp<<endl;
            if(temp.find(b)!=-1){
                return count;
            }
        }
        return -1;
    }
};
```

Output:



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

The screenshot shows the LeetCode interface. The top navigation bar includes the LeetCode logo, a 'Problem List' button, a 'Premium' badge, and a user profile icon. The main content area is divided into two panels. The left panel, titled 'Submissions', shows a list of submissions for problem 459, 'Repeated Substring Pattern'. The top submission is marked 'Accepted' and was made 'a few seconds ago' using C++. Below it, a submission is marked 'Wrong Answer' and was made '2 minutes ago' using C++. The right panel, titled 'Close', displays the C++ code for the 'Accepted' submission. The code defines a class 'Solution' with a public method 'repeatedStringMatch' that takes two strings 'a' and 'b' and returns an integer. The code uses a while loop to repeatedly append string 'a' to a temporary string 'temp' until its length is greater than or equal to string 'b'. It then checks if 'b' is a substring of 'temp' using the 'find' method. If it is, it returns the count of 'a' in 'temp'; otherwise, it returns -1.

```
class Solution {
public:
    int repeatedStringMatch(string a, string b) {
        int la=a.size();
        int lb=b.size();
        int k=(ceil(lb/la))+2;
        int count=0;
        string temp="";
        while(k--){
            temp+=a;
            count++;
            // cout<<temp<<endl;
            if(temp.find(b)!=-1){
                return count;
            }
        }
        return -1;
    };
};
```

At the bottom of the right panel, there are buttons for 'Console', 'Run', and 'Submit'.