Description | Editorial | Solutions (256) | Submissions

2825. Make String a Subsequence Using Cyclic Increments

Hint

Medium 168 5

Companies

You are given two **0-indexed** strings `str1` and `str2`.

In an operation, you select a **set** of indices in `str1`, and for each index `i` in the set, increment `str1[i]` to the next character **cyclically**. That is `'a'` becomes `'b'`, `'b'` becomes `'c'`, and so on, and `'z'` becomes `'a'`.

Return `true` *if it is possible to make* `str2` *a subsequence of* `str1` *by performing the operation* ***at most once****, and* `false` *otherwise.*

Note: A subsequence of a string is a new string that is formed from the original string by deleting some (possibly none) of the characters without disturbing the relative positions of the remaining characters.

Example 1:

```
Input: str1 = "abc", str2 = "ad"
Output: true
Explanation: Select index 2 in str1.
Increment str1[2] to become 'd'.
Hence, str1 becomes "abd" and str2 is now a subsequence.
Therefore, true is returned.
```

Example 2:

```
Input: str1 = "zc", str2 = "ad"
Output: true
Explanation: Select indices 0 and 1 in str1.
```

Java | Auto

```java
class Solution {
    public boolean canMakeSubsequence(String str1, String str2) {

        if(str2.length()>str1.length()) return false;

        int i =0,j=0;

        while(i<str1.length())
        {

            char ch = str1.charAt(i);
            char ch2 =str2.charAt(j);

            if(ch=='z' && ch2=='a') j++;

            else if(ch2-ch==1 || ch2-ch==0){
                j++;
            }
```

Ln 27, Col 2

Testcase | Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

str1 =
"abc"

str2 =
"ad"

Output

Console | Run | Submit

Description | Editorial | Solutions (2.6K) | Submissions

168. Excel Sheet Column Title

Easy 4.8K 667

Companies

Given an integer `columnNumber`, return *its corresponding column title as it appears in an Excel sheet.*

For example:

```
A -> 1
B -> 2
C -> 3
...
Z -> 26
AA -> 27
AB -> 28
...
```

Example 1:

```
Input: columnNumber = 1
Output: "A"
```

Example 2:

```
Input: columnNumber = 28
Output: "AB"
```

Example 3:

```
Input: columnNumber = 701
Output: "ZY"
```

Java | Auto

```java
public class Solution {
    public String convertToTitle(int columnNumber) {
        StringBuilder out = new StringBuilder();
        while (columnNumber > 0) {
            out.insert(0, (char) ('A' + (columnNumber - 1) % 26));
            columnNumber = (columnNumber - 1) / 26;
        }
        return out.toString();
    }
}
```

Ln 10, Col 2

Testcase | Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

columnNumber =
1

Output

"A"

Expected

Console | Run | Submit