Leetcode
Daily Challenge

T.C. o(n)
S.C. o(1)

02/03/2022

problem

Is Subsequence

pre-requisites

2 pointer

difficulty

Easy

est. time

10-15 min

can be asked in...

49%
Accuracy

Statement

Description

- Given two strings s and t, return true if s is a subsequence of t, or false otherwise.

I/P	O/P
s = "abc", t = "ahbgdc"	true

Subsequence->

- string s = "abc", can be formed removing by o or more chars of t = "ahbgdc", so s is subsequence.
- from t = "ahbgdc" removing 'h','g','d' gives s = "abc".

Intuition

- You can obviously go with Recursion but here we don't want all the possibility (count) of subsequece s in string t.
- We just need to see if 1 subsequence exists or not.

0	1	2	3	4	5	idx
"a	h	b	g	d	c"	t
"a	-	b	-	-	c"	s

- from above pic, it is clear that if we can have a variable that keeps track of indices in S whose value we found in T & finally if that variable reaches end of S, we say S is subsequence of T.
- So idea is we use 1 variable to iterate t & 1 var. to iterate s & what is this technique called

2 Pointer

Algorithm

- int i= 0 (to iterate t)
- int j = 0 (to iterate s)
- iterate over 't'
  `if(t[i] == s[j]) j++;`
- `if(j == s.size()) return 1`
  `else return 0`

Dry Run

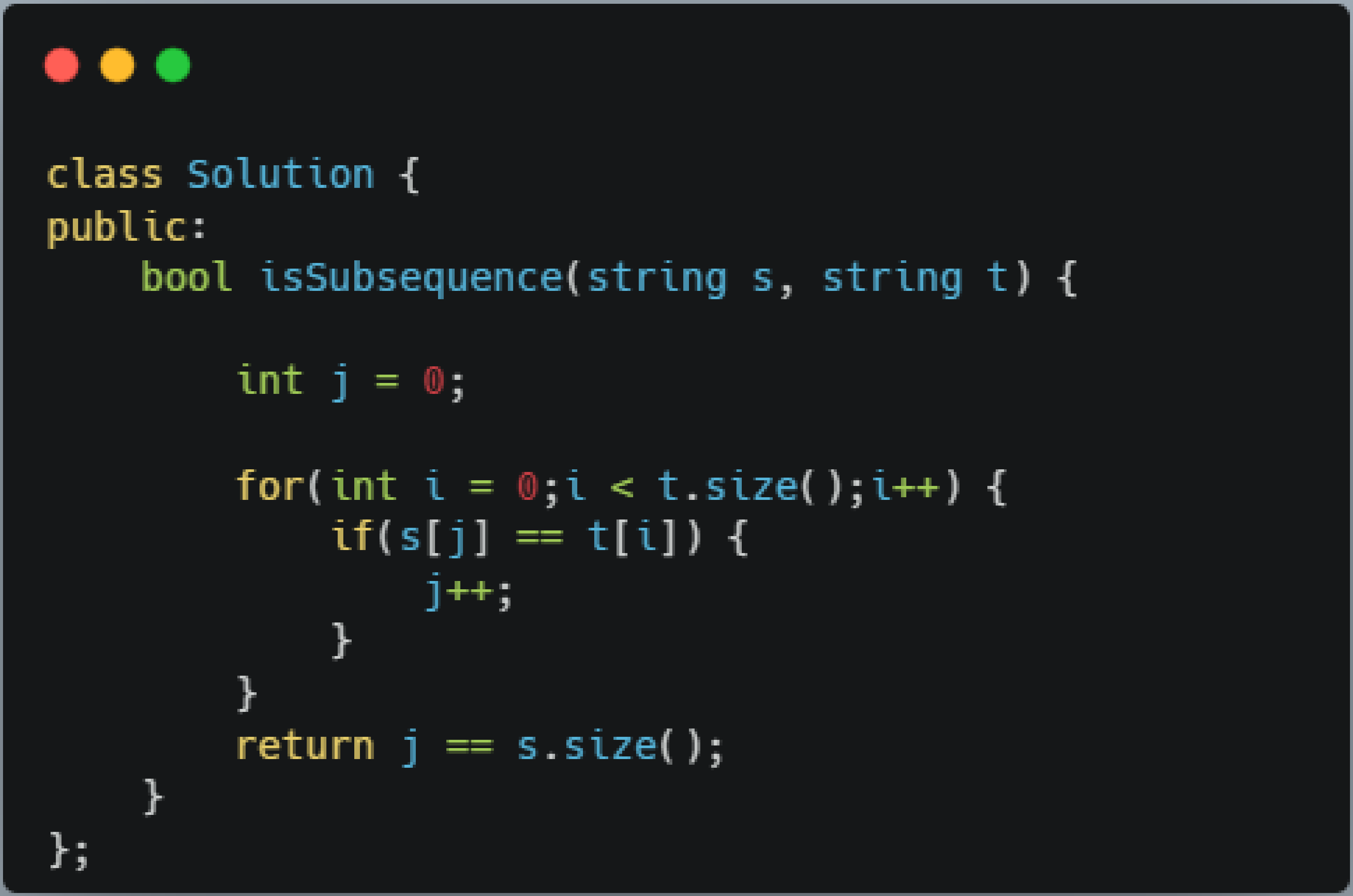
```
t = "ahbgdc"
s = "abc"
```

i	j	t[i]	s[j]	comment
0	0	'a'	'a'	a==a; j++, i++
1	1	'h'	'b'	h!=b; i++
2	1	'b'	'b'	b==b; j++, i++
3	2	'g'	'c'	g!=c; i++
4	2	'd'	'c'	d!=c; i++
5	2	'c'	'c'	c==c; j++, i++
6	3	-	-	final state

```
here j == s.size() i.e.
     3 == 3, return 1
```

j reaches end of s, so s is subseq. of t

we found all chars of s in t in order



```
class Solution {
public:
    bool isSubsequence(string s, string t) {

        int j = 0;

        for(int i = 0;i < t.size();i++) {
            if(s[j] == t[i]) {
                j++;
            }
        }
        return j == s.size();
    }
};
```

Leave a Like

Comment if you love posts like this, will motivate me to make posts like these

Share, with friends