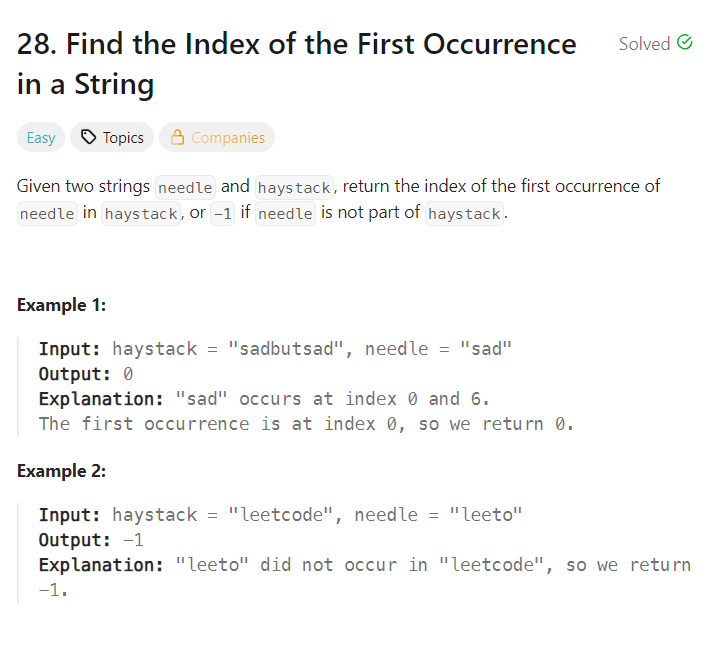
So here is the question:  


I was thinking of a Brute Force approach but the time complexities were too high so I thought it might not run on leetcode so why bother with the brute force approach. I started racking my brain for options as to how I could solve this, I thought of two approaches, the Two Pointer approach and the Sliding window approach.

The sliding window approach seemed the most perfect way as we can have the window size instantly and then we just keep moving the window. But since I have never tried Sliding window approach, I thought of first getting the solution using Two pointers.

I couldn’t exactly write the solution using two pointers, there were a few things missing but after a few optimizations I had this solution.

Have a outer loop with variable i, which runs until <= haystack.length() – needle.length() as we do not want to go out of bounds.  
Run the inner loop to check for every character from i to needle.length();

If we find a match we return i; If not we return -1;

Here is the code:

**class Solution {**

**public int strStr(String haystack, String needle) {**

**for(int i=0; i<=haystack.length() - needle.length(); i++ ){**

**int j=0;**

**while(j < needle.length() && haystack.charAt(i+j) == needle.charAt(j) ){**

**j++;**

**}**

**if(j == needle.length()){**

**return i;**

**}**

**}**

**return -1;**

**}**

**}**

Here is the **sliding window approach** that was pretty straightforward, we use the substring method to check with the needle:

**class Solution {**

**public int strStr(String haystack, String needle) {**

**int windowSize = needle.length();**

**for(int i=0; i<= haystack.length() - needle.length(); i++){**

**String window = haystack.substring(i, i+windowSize);**

**if(window.equals(needle)){**

**return i;**

**}**

**}**

**return -1;**

**}**

**}**

I found that the Sliding window approach ran with lower time complexity than the Two pointer approach. Although it might just be a leetcode thing that time complexity keeps changing with number of submissions.