Meta

Above and Beyond Computer Science

Workshop 2: Strings & Arrays - Student Handout

FACEBOOK IN A STRING (EASY)

QUESTION:
Given a string, determine if there is a sequential subsequence of characters that spell the word “facebook”.

Constraints
• The input string could be empty or null.
• The input string will only contains letters a-z and A-Z.
• The input string will be less than 10000 characters in length.

Example

"fffaaccccebboook" – true
"fffaaccccebbok" – false
"fffaacccbook" – false
"i wonder if the facebook would be more popular if it had that prefix" – true
"i found a book about how to wash your face" - false

Meta

FACEBOOK IN A STRING (EASY)

A SOLUTION:
Loop over the input string while keeping a pointer to the character in the “facebook” string. Once a character in the input string matches the current character in the facebook string, increment the pointer. Once the pointer has traversed the whole facebook string, we know there is a subsequence.

```
def fbSubsequence(input):
  facebook = "facebook"
  fb_ptr = 0

  for i in range(len(input)):
    if input[i] == facebook[fb_ptr]:
      fb_ptr+=1
    if fb_ptr == len(facebook):
      return True

  return False

print(fbSubsequence("fffaaccccebbooook")) # true
print(fbSubsequence("fffaaccccebbok")) # false
print(fbSubsequence("fffaacccbook")) # false
print(fbSubsequence("i wonder if the facebook would be more
popular if it had that prefix")) # true
print(fbSubsequence("i found a book about how to wash your face"))
# false
```

Runtime:
O(n) (length of input)

Space Complexity:
O(1)

MAXIMUM SUBARRAY SUM (MEDIUM)

QUESTION:
We define subsequence as any subset of an array. We define a subarray as a contiguous subsequence in an array.

Given an array, find the maximum possible sum among all possible subarrays..

Example

[1, 2, 3 ,4] = 10
[2, -1, 2, 3, 4, -5] = 10
[2, -1] = 2
[-3, -2,- 1] = -1

MAXIMUM SUBARRAY SUM (MEDIUM)

A SOLUTION:

This is called **Kadane's algorithm**. For this question, For every step, it computes the largest sum subarray ending at index i. This is current_sum. It also computes the subarray anywhere in arr[0...i], this value is stored in max_sum.

```
def maximumSubarray(arr):
    max_sum = 0
    current_sum = 0

    for i in range(len(arr)):
        current_sum = current_sum + arr[i]
        if (max_sum < current_sum):
            max_sum = current_sum
        if (current_sum < 0):
            current_sum = 0

    # handle negative maximum subarray
    if max_sum <= 0:
        return max(arr)
    else:
        return max_sum

print(maximumSubarray([1,2,3,4])) # 10
print(maximumSubarray([2,-1])) # 2
print(maximumSubarray([2,-1,2,3,4,-5])) # 10
print(maximumSubarray([-3, -2,-1])) # -1
```

Runtime:

O(n)

Space Complexity:

O(1)

Meta

Soft Skills Checklist

Working the Clock Did the person…	☐ Spend 5min before writing any code to communicate proactive and design their algorithm? ☐ Spend 10min coding, including talking through their solution and handling any mistakes? ☐ Spend 2-3min to test their solution?
Communicate Proactively Did the person…	☐ Repeat the question and rephrase in their own words? ☐ Assume all of the information that is given is necessary to solve the problem? ☐ Ask questions to clarify the scope and intention of the problem, validate or state assumptions, or resolve edge cases.
Designing an Algorithm Did the person…	Stay tuned!
Writing Code at the Whiteboard Did the person…	Stay tuned!
Talking Through Code/Solution Did the person…	Stay tuned!
Handling Mistakes Did the person…	Stay tuned!
Test Your Code Did the person…	Stay tuned!
Increasing Coding Speed Did the person…	Stay tuned!