

1. 단어중복없는 가장 긴 문자열

(Longest Substring Without Repeating Characters)

2. 최대 2 개의 고유 문자가있는 가장 긴 부분 문자열

(Longest Substring with At Most Two Distinct Characters)

3. 문자열에서 모든 ана그램 찾기

(FindAllAnagramInAString)

단어중복없는 가장 긴 문자열 (Longest Substring Without Repeating Characters)

설명

String *s*가 주어집니다.
중복이 없는 가장 긴 substring 문자열을 리턴하세요

입출력

Input: *s* = "abcabcd"

Output: 4

Explanation: "abcd"

Input: *s* = "pwwkea"

Output: 4

Explanation: "wkea"

Input: *s* = "aaaaa"

Output: 1

Explanation: "a"

Input: *s* = ""

Output: 0

단어중복없는 가장 긴 문자열 (Longest Substring Without Repeating Characters)

문제 Format

```
class Solution {  
    public int solve(String s) { }  
}
```

제한사항

$0 \leq s.length \leq 5 * 10^4$
s consists of English letters

최대 2 개의 고유 문자가있는 가장 긴 부분 문자열 (Longest Substring with At Most Two Distinct Characters)

설명

String s 가 주어집니다.
최대 두 개의 고유 문자 를 포함하는 가장 긴 부분 문자열의 길이를 리턴하세요

입출력

Input: $s = \text{"eceba"}$

Output: 3

Explanation: "ece"

Input: $s = \text{"ccaabbbb"}$

Output: 5

Explanation: "aabbbb"

문제 Format

```
class Solution {  
    public int solve(String s) {}  
}
```

제한사항

$1 \leq s.length \leq 10^4$
 s consists of English letters.

문자열에서 모든 아나그램 찾기 (FindAllAnagramInAString)

설명

두 개의 문자열이 주어집니다. String s, p
p의 문자열이 s의 문자열에 특정인덱스에서부터 모든 아나그램 찾아서 인덱스를 리턴하세요

입출력

Input: s = "bacdgabcda", p = "abcd"

Output: [0,5,6]

Explanation:

0번 인덱스 bacd

5번 인덱스 abcd

6번 인덱스 bcda

p=abcd는 s에 특정 인덱스에 대입했을때,
아나그램(순서는 상관없이 단어일치) 되는 시점을 반환합니다.

문제 Format

```
class Solution {  
    public List<Integer> solve(String s, String p) { }  
}
```

제한사항

$1 \leq s.length, p.length \leq 3 * 10^4$
s and p consist of lowercase English letters.