Given two strings s and p, return an array of all the start indices of p's anagrams in s. You may return the answer in **any order**.

**Example 1:**

**Input:** s = "cbaebabacd", p = "abc"

**Output:** [0,6]

**Explanation:**

The substring with start index = 0 is "cba", which is an anagram of "abc".

The substring with start index = 6 is "bac", which is an anagram of "abc".

**Example 2:**

**Input:** s = "abab", p = "ab"

**Output:** [0,1,2]

**Explanation:**

The substring with start index = 0 is "ab", which is an anagram of "ab".

The substring with start index = 1 is "ba", which is an anagram of "ab".

The substring with start index = 2 is "ab", which is an anagram of "ab".

**Constraints:**

* 1 <= s.length, p.length <= 3 \* 104
* s and p consist of lowercase English letters.

Progrm:

from collections import Counter

class Solution:

    def findAnagrams(self, s: str, p: str) -> List[int]:

        def is\_anagram(c1, c2):

            for ch in "abcdefghijklmnopqrstuvwxyz":

                if c1[ch] != c2[ch]:

                    return False

            return True

        p\_len = len(p)

        s\_len = len(s)

        if p\_len > s\_len:

            return []

        p\_counter = Counter(p)

        s\_counter = Counter(s[:p\_len])

        result = []

        if is\_anagram(p\_counter, s\_counter):

            result.append(0)

        for i in range(1, s\_len - p\_len + 1):

            if s[i-1] != s[i+p\_len-1]:

                s\_counter[s[i-1]] -= 1

                s\_counter[s[i+p\_len-1]] += 1

            if is\_anagram(p\_counter, s\_counter):

                result.append(i)

        return result

Program 2:

from collections import Counter

from typing import List

class Solution:

    def findAnagrams(self, s: str, p: str) -> List[int]:

        res = []

        p\_count = Counter(p)

        s\_count = Counter(s[:len(p)])

        if s\_count == p\_count:

            res.append(0)

        for i in range(len(p), len(s)):

            s\_count[s[i]] += 1

            s\_count[s[i - len(p)]] -= 1

            if s\_count[s[i - len(p)]] == 0:

                del s\_count[s[i - len(p)]]

            if s\_count == p\_count:

                res.append(i - len(p) + 1)

        return res