**Problem #438: Find All Anagrams in a String**

<https://leetcode.com/problems/find-all-anagrams-in-a-string/>

**My Solution:**

1. If length of p is greater than length of s return empty list.
2. Let n be the length of s.
3. Let k be the length of p.
4. Initialize dict\_p to be an empty dictionary.
5. Create a frequency dictionary for p in dict\_p.
6. Create a frequency distribution for s in dict\_s.
7. Initialize index\_lsit to an empty list.
8. If dict\_s and dict\_p are equal, then append 0 to index\_List.
9. Traverse through s in range n – k. If the key at index (I + k )is not in the dict\_s keys, then add the element to the dictionary and set its value to 0.

Otherwise, if it is in the dictionary already, then increment the value by 1.

Decrement the value of key at index I by 1. If it becomes 0, then pop this value from the dictionary. If dict\_s and dict\_p are equal then append (I + 1) to index\_list.

1. Return index\_list.

**Explanation:**

In step 9, we are not creating the whole dictionary of dict\_s everytime. We are using a sliding window approach. We are considering from I to (i+k), , then (I + 1) to (I + k + 1) and so on. So the second time, we just update the existing dictionary with (I + k + 1) and remove the effect of element at index i.

from collections import Counter

class Solution:

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

'''

n = len(s)

k = len(p)

dict\_p = Counter(p)

start\_index = []

for i in range(n - k + 1):

if Counter(s[i: i + k]) == dict\_p:

start\_index.append(i)

return start\_index

'''

if len(p) > len(s):

return []

n = len(s)

k = len(p)

dict\_p = {}

for i in range(k):

if p[i] not in dict\_p.keys():

dict\_p[p[i]] = 1

else:

dict\_p[p[i]] += 1

print("dict\_p = ", dict\_p)

dict\_s = {}

for i in range(k):

if s[i] not in dict\_s.keys():

dict\_s[s[i]] = 1

else:

dict\_s[s[i]] += 1

print("After putting s[0: k] dict\_s = ", dict\_s)

index\_list = []

if dict\_s == dict\_p:

index\_list.append(0)

for i in range(n - k):

print("i = ", i, "s[i + k] = ", s[i + k])

if s[i + k] not in dict\_s.keys():

dict\_s[s[i + k]] = 1

else:

dict\_s[s[i + k]] += 1

#print("i = ", i, "s[i] = ", s[i], "dict\_s = ", dict\_s)

dict\_s[s[i]] -= 1

if dict\_s[s[i]] == 0:

dict\_s.pop(s[i])

if dict\_s == dict\_p:

index\_list.append(i + 1)

#print("loop end dict\_s = ", dict\_s, "index\_list = ", index\_list)

return index\_list