Longest Substring With At Most K Unique Characters

k = 3

b d a b a a c b c a d b c a a

atmost 3

k = 3

a b a a

ans : abaa

k = 3

j

b d a b a a c b c a d b c a a

i

b - 1

0 ans = ~~1~~ ~~2~~ ~~3~~ 4 ~~5~~ ~~6~~ ~~7~~ 8

a - 2

c - 1

map.size() <= k

-> ans updation.

(i) aquire till invalid.

(ii) release to be valid again.

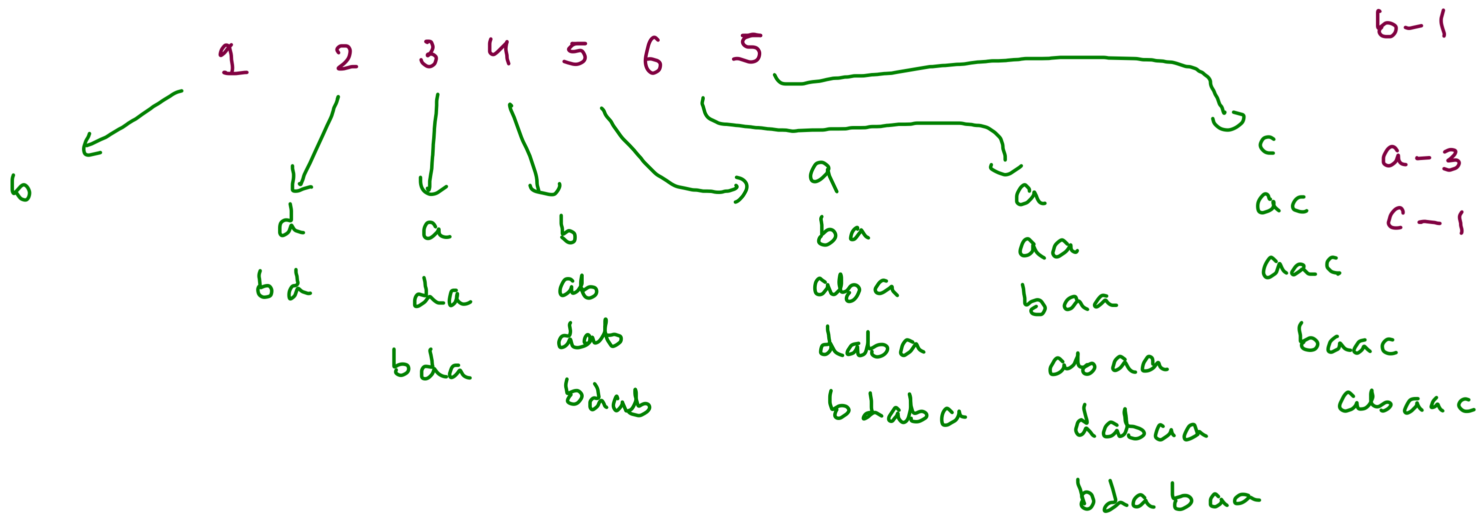
K = 3

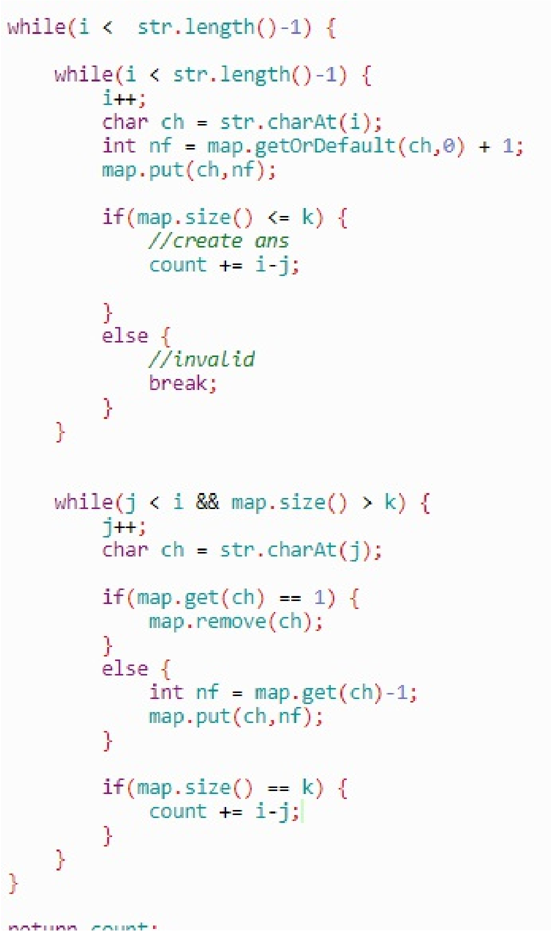
count

j

b d a b a a c b c a d b c a a

i





```
while(i <  str.length()-1) {

    while(i < str.length()-1) {
        i++;
        char ch = str.charAt(i);
        int nf = map.getOrDefault(ch,0) + 1;
        map.put(ch,nf);

        if(map.size() <= k) {
            //create ans
            count += i-j;

        }
        else {
            //invalid
            break;
        }
    }


    while(j < i && map.size() > k) {
        j++;
        char ch = str.charAt(j);

        if(map.get(ch) == 1) {
            map.remove(ch);
        }
        else {
            int nf = map.get(ch)-1;
            map.put(ch,nf);
        }

        if(map.size() == k) {
            count += i-j;
        }
    }
}

return count;
```

k = 2

j

a a b a

i

a

aa
a

aab
ab
b

aaba
aba
ba
a

c = 1 + 2 + 3 + 4 = 10

a - 3
b - 1

Valid Anagram

d a b b a c c d d

↳

a - 2

b - 2

c - 2

d - 3

a a b b c c d d d

↳

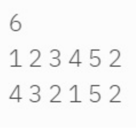
a - 2

b - 2

c - 2

d - 3

Find Anagram Mappings



```
6
1 2 3 4 5 2
4 3 2 1 5 2
```

ans :	4	0	1	2	3	5
a1	1_0	2_1	3_2	2_3	3_4	2_5
a2	2_0	3_1	2_2	3_3	1_4	2_5

←

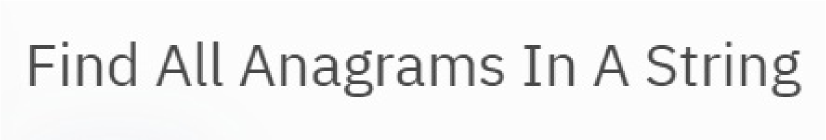
ele vs positions

int vs AL<int>

2 -> [~~5~~, ~~2~~, ~~0~~]

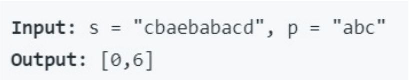
1 -> [~~4~~]

3 -> [~~3~~, ~~1~~]

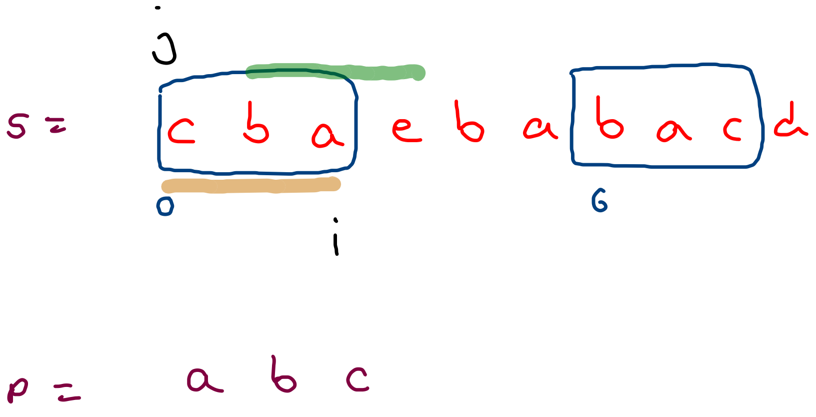


Find All Anagrams In A String

Sliding window



```
Input: s = "cbaebabacd", p = "abc"
Output: [0,6]
```



[0,6]

O(n)

j

b	a	c	d	a	b	c	a	d	b	c	a	a	b	c
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
				i										

4 9 10 11

compare : 26 (constant)

wm b-1
a-2
c-1

pm: a-2
b-1
c-1

p = abac

j

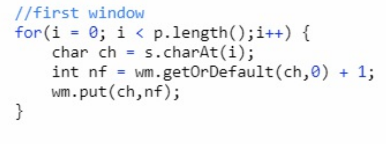
b a c d a b c a d b c a a b c

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

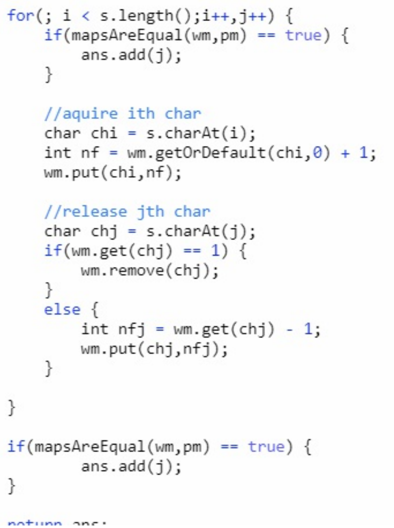
i

wm : a - 2
c - 1
b - 1

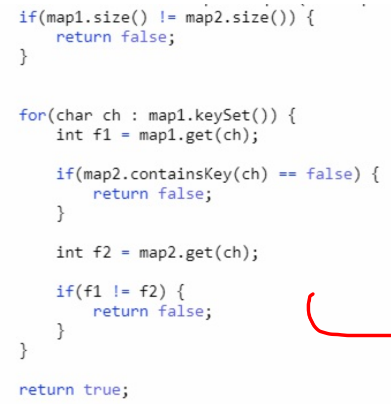
pm : a - 2
b - 1
c - 1



```
//first window
for(i = 0; i < p.length();i++) {
    char ch = s.charAt(i);
    int nf = wm.getOrDefault(ch,0) + 1;
    wm.put(ch,nf);
}
```



```
for(; i < s.length();i++,j++) {
    if(mapsAreEqual(wm,pm) == true) {
        ans.add(j);
    }

    //aquire ith char
    char chi = s.charAt(i);
    int nf = wm.getOrDefault(chi,0) + 1;
    wm.put(chi,nf);

    //release jth char
    char chj = s.charAt(j);
    if(wm.get(chj) == 1) {
        wm.remove(chj);
    }
    else {
        int nfj = wm.get(chj) - 1;
        wm.put(chj,nfj);
    }

}

if(mapsAreEqual(wm,pm) == true) {
        ans.add(j);
}

return ans;
```



```
if(map1.size() != map2.size()) {
    return false;
}


for(char ch : map1.keySet()) {
    int f1 = map1.get(ch);

    if(map2.containsKey(ch) == false) {
        return false;
    }

    int f2 = map2.get(ch);

    if(f1 != f2) {
        return false;
    }
}

return true;
```

p = abac

ans = 4, 9, 10, 1

→ constant time

O(26)