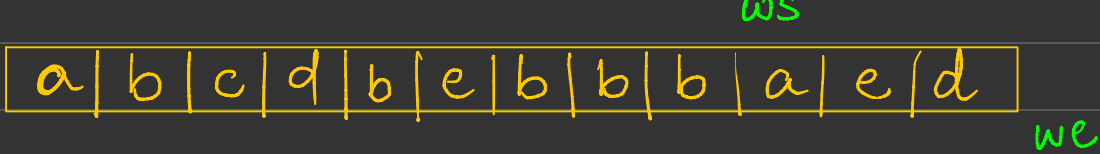
Longest substring with no more than k dist. characters

Ex: s: "abcdaebbbaed" K=3



maxlen: ~~0 1 2 3 4 5 6~~ 7

map: { d:1 e:1 a:1 }

maxlen = max(maxlen, we - ws + 1)

Alice & Bob: siblings
↳ Indian but living in abroad with their parent
↳ Diwali festival in their school.
↳ sweets distribute
↳ distributes sweets on this Diwali
↳ allowed ✓ but there are some conditions

N sweets stalls in the market.
→ You can start with any stall
→ You can not buy more than K different sweets

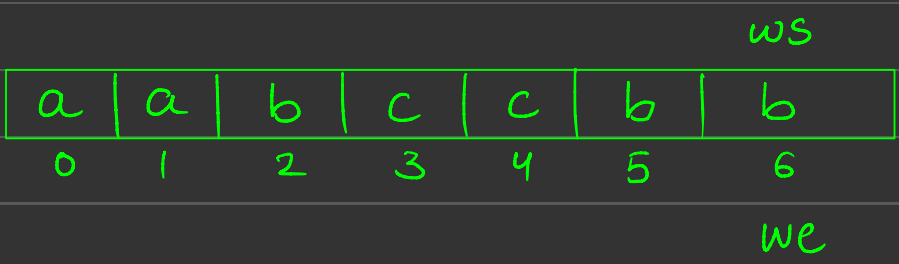
```
def findLength(s, k):
    maxlen = 0
    ws = 0
    freqMap = {}

    for we in range(len(s)):
        curr_char = s[we]
        freqMap[currchar] = freqmap.get(
                                currchar, 0) + 1

    while len(freqmap) > k:
        lftchar = s[ws]
        freqmap[lftchar] -= 1
        if freqMap[lftchar] == 0:
            del freqmap[lftchar]
        ws += 1
    maxlen = max(maxlen, we - ws + 1)
return maxlen
```

map<char, int> fmap;

longest Substring without Repeating characters



maxlen = ~~0~~ ~~1~~ ~~2~~ 3

{ a:1 b:6 c:4 }

maxlen = max(maxlen, we - ws + 1)

```
def findlen(s):
    mxlen = 0
    ws = 0
    imap = {}

    for we in range(len(s)):
        char = s[we]
        if char in imap:
            prevIndex = imap[char]
            if prevIndex >= ws:
                ws = prevIndex + 1

        imap[char] = we
        mxlen = max(mxlen, we - ws + 1)

    return mxlen
```

ws = max(ws, prevIndex + 1)