Longest Substring without repeating characters

Leetcode - 1

Input: Given a string

Output: Length of the longest substring without repeating characters.

Ex: i/p: abcabcbb
o/p: 3

i/p: bbbb
o/p: 1

i/p: nfpdmpi
o/p: 5

i/p: aslj1j
o/p: 4

Approach: O(n) - linear time

* Created an array of 256 characters with initial values as -1
* This array is updated with the index of the character that is processed.

So if 'b' occurs at index '2' in a string, then, the array ~~with~~ ~~with~~ ~~index~~ whose character 'b' is equal to 2.

* Now we have to find the length of the longest sub string.

So cur_len } two variables are initialized to '1'
max_len }

* process the 0th element of the string and create a for loop from 1 to n.

* Now for each character of a string there are three cases to be handled:

(a) Has the character ~~pass~~ seen before?

ans: Check in the array. If the array has value -1. It is a new character.

So, increment the cur_len by 1 and add the ~~position of~~ index of the character in the visited array

(b) Yes it is already seen

Does the character present in the current_len (NCRS)

Current non repeating character string (NCRS)

↓

Yes it is present.

Then the cur_len is updated.

Ex: b a l b

cur_len 3 ↑ new character

b is already in cur_len so it is updated to

b a l b

cur_len

Does the character is not present in current len (NCRS)

↓

NO

Then good to go, just increase cur_len by 1

b a b l b

cur_len ↑ new character

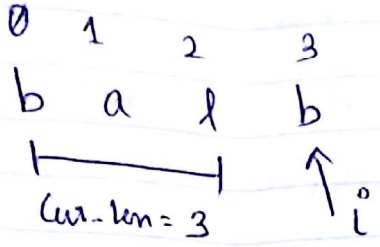
b a b l b

cur_len

LC-2

How to find if the ~~string~~ character is present in current WCRS?

Ans:



// Here prev_index = 0 that is the index at which b is occurred (this stored in visited array)

```
if i - cur_len > prev_index :
    // the character is not
       in cur_len
else:
    // the character in cur_len
       so update cur_len

    cur_len = i - prev_index
```

* Once this is done, check cur_len & max_len return max_len.

What I have learned here?

* String problems:

(a) Create a visited array of 256 size with values initialized to -1

(b) Use this visited array to track the index of a character

(c) Check if the ~~character~~ char is part of the current longest sequence

i - cur_len ≤ prev_index

(d) If not

~~i - prev~~

i - cur_len > prev_index

(e) To change the sequence length

cur_len = i - prev_index

[check examples]