S= " abcabcbb"

Find length of longest substring without repeating characters.

I have to cross reference, so, I will use a hashmap to store the values that I have seen before.

First create a hashmap to cross reference.

hashmap= { }

Then initialize max_len and left pointer to 0.

left=0

max_len=0

Iterate through the string.

for i in range (len(s)):-

Check if the character is in the hashmap:-

if (s[i] in hashmap):-

Check if the character I am iterating right now, if that character's index is bigger than the left, meaning I found a duplicate of the character I am iterating.

Combining both of them, we get,

$\text{If } (\underbrace{s[i] \text{ in hashmap}}_{\text{Found a duplicate.}} \text{ and } \underbrace{(\text{hashmap}[s[i]] \geq \text{left})}_{\text{Index of duplicate in original array is greater than the starting of current string)}}):-$

$\underbrace{\text{left} = \text{hashmap}[s[i]]+1.}_{}$

Start counting new substring and move pointer to next element.

After cross checking duplicates, we have to put characters in hashmap too.

The keys are going to be the characters

The values are going to be the index.

hashmap[s[i]]=i

Then we have to update the value of max_len.

max_len=max (max_len, i-left +1).

return max_len.