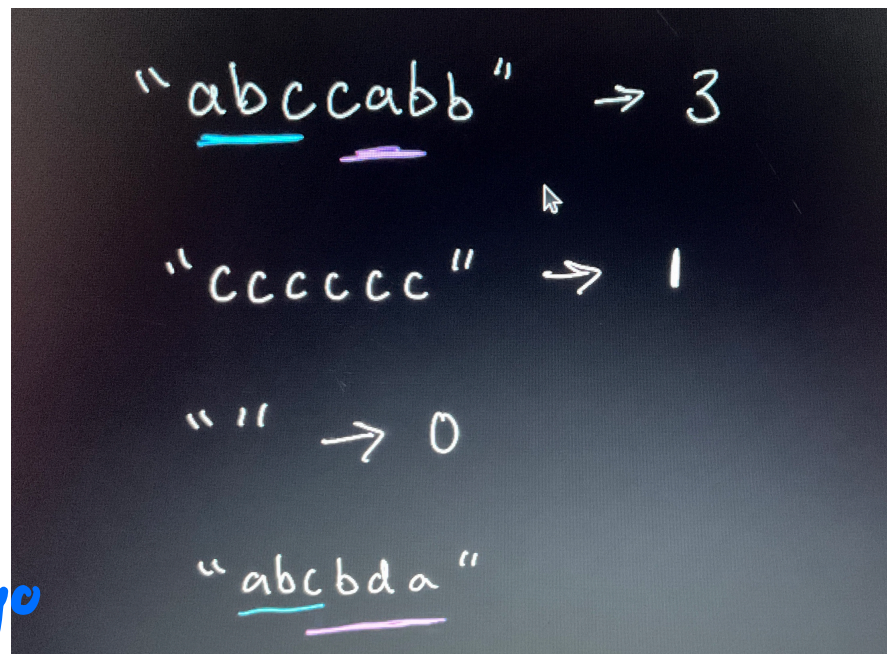


Cadenas

Encontrar la subcadena más larga dentro de una cadena.



Pseudocódigo

```
func length_of_longest_substring(string: str) → int:  
    if len(string) ≤ 1:  
        return len(string)
```

Longest = \emptyset

```
for left in range(len(string)):
```

seen_chars = {}

current_length = \emptyset

```
for right in range(left, len(string)):
```

current_char = string[right]

if not (current_char in seen_chars.keys()):

current_length += 1

seen_chars[current_char] = True

Longest = max(Longest, current_length)

else:

break

return longest

Optimized

func length-longest-substring(string: str) -> int:

· if len(string) <= 1:
 return len(string)

seen_char = {}

left, longest = 0, 0

for right in range(len(string)):

current_char = string[right]:

prev_seen_char = -1

if current_char in seen_char.keys():

prev_seen_char = seen_char[current_char]

if prev_seen_char >= left:

left = prev_seen_char + 1

seen_char[current_char] = right

longest = max(longest, (right - left + 1))

return longest