Algorithm Find the longest palindromic sequence in a given string

Ensure: Zero Based Indexing

1: **function** LONGEST_PALINDROME(str)
    ▷ If $dp[i][j]$ is *true*, $str[i...j]$ is a palindrome.
2:    **if** str is empty **then**
3:        **return** ""
4:    $size \leftarrow str.len$
5:    $dp \leftarrow \{false\}$
6:    **for** $i = size - 1 : 0$ **do**
7:        **for** $j = i : size - 1$ **do**
8:            $length \leftarrow j - i + 1$
9:            **if** $str[i] == str[j]$ **then**
10:                **if** $length \leq 2$ **then**
11:                    $dp[i][j] \leftarrow true$
12:                **else**
13:                    $dp[i][j] \leftarrow dp[i + 1][j - 1]$
14:    $start \leftarrow 0$
15:    $max\_length \leftarrow 1$
16:    **for** $i = 0 : size - 1$ **do**
17:        **for** $j = i : size - 1$ **do**
18:            **if** $dp[i][j] == false$ **then**
19:                *continue*
20:            $length \leftarrow j - i + 1$
21:            **if** $length > max\_length$ **then**
22:                $start \leftarrow i$
23:                $max\_length \leftarrow length$
24:    **return** $str.substring(start, max\_length)$