Let $A = (a_1, a_2, \dots, a_n)$ be an array of characters.

Let $1 <= i <= j <= n$

Base cases:

$i = j$ → we return 1 since by definition a sequence of length 1 is a palindrome

$j = i + 1 \; and \; a_i = a_j$ → we return 2 (minimum size of an even length palindrome is 2)

Assume $i \neq j \; and \; j \neq i + 1$

if $a_i = a_j$ then we will count 2 and recursively check $a_{i+1}, a_{i+2}, \dots, a_{j-1}$

if $a_i \neq a_j$ then the only two options are to look in $a_i, a_{i+1}, \dots, a_{j-1}$ and

in $a_{i+1}, a_{i+2}, \dots, a_j$

therefore, we define $f: \mathbb{N} \; X \; \mathbb{N} \to \mathbb{N}$ by:

$$f(i,j) = \begin{cases} 1, & i = j \\ 2, & j = i + 1 \; and \; A[i] = A[j] \\ 2 + (f(i+1, j-1), & A[i] = A[j] \\ \max\big(f(i+1, j), f(i, j-1)\big), & otherwise \end{cases}$$

if you write a recursion tree for f then you will see that there're more than one computation for some inputs. We can initialize an $n * n$ matrix at treat an entry $[i, j]$ as $f(i, j)$

just translate f as defined above to an entry of A.

You can see the source code for iterative implementation

correctness is easy in this case. Use induction on n