Find the *longest subsequence palindrome* in a given array `a`. The *longest subsequence palindrome* of array `a` is a subsequence of indices $i_1 < i_2 < ... < i_k$, where $a_{i_1}a_{i_2}...a_{i_k}$ is a palindrome.

Example

- For `a = [1, 2, 4, 1]`, the output should be `longestSubsequencePalindrome(a) = 3`.

  The *longest subsequence palindrome* here is either `1, 2, 1` or `1, 4, 1`, both of which have a length of `3`.

- For `a = [1, 2, 3]`, the output should be `longestSubsequencePalindrome(a) = 1`.

Input/Output

- **[time limit] 4000ms (py3)**
- **[input] array.integer a**

  Guaranteed constraints:
  $1 \leq a.length \leq 10^3$,
  $1 \leq a[i] \leq 10^9$.

- **[output] integer**

  The length of the *longest subsequence palindrome* in `a`.