```python
def prefix_palindrome(s):
    for k in range(len(s), 0, -1):
        # check if s[0:k] is a pal.
        is_pal = True
        for j in range(k):
            if s[j] != s[k-1-j]:
                is_pal = False
                break  # s[0:k] is not a pal
        if is_pal:
            return k  # s[0:k] is a pal
```

1. Worst-case upper bound

For a fixed iteration of the outer loop (i.e., fixed value of k), there are at most k iterations. So the
are at most $k$ iterations. So the cost is at most $k$ steps (1 step per iteration).

Let $n$ be the length of the input string. For the outer loop, the # of iterations is at most $n$ ($k=n,n-1,...,2$)

So the total cost is \( \text{AT MOST} \)

\[
\sum_{k=1}^{n} k = \frac{n(n+1)}{2} .
\]

So the $WC \in O(n^2)$.

\( \text{2) Worst-case lower bound.} \)

$WC \in \Omega(n^2)$ ("matching lower bound")

We will prove there exists an input family (one input per $n \in \mathbb{Z}^+$).
input family (one input per $n \in \mathbb{Z}$), whose runtime is $\Omega(n^2)$.

<table>
<thead>
<tr>
<th>k</th>
<th># iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

$\sum_{i=1}^{n} i = \Omega(n^2)$

e.g. \[abcdefgaba\]

$\sum_{i=1}^{n} i \in \Omega(n^2)$

Average-case analysis
return False

Goal: Find out how well find_one does "on average"

Define the function

\[ \text{AVG}(n) = \frac{\text{average}}{\text{find-one}} \]

\[ \text{average of \{runtime of find_one(L)\}} \]

\[ L \text{ has len } n \}

* AVG(n) depends on what inputs we consider.

For this example, we pick these inputs:

- For each \( n \in \mathbb{Z}^+ \), the lists that are permutations of \( \{1, 2, \ldots, n\} \).

E.g., the lists of length...

\[ n=1 \quad \{1\} \]

\[ n=2 \quad \{1, 2\}, \{2, 1\} \]
\[ n = 3 \quad [1,2,3], [1,3,2], \]
\[ [2,1,3], [2,3,1], \]
\[ [3,1,2], [3,2,1] \]

\[ \forall n \in \mathbb{Z}^+, \text{ the number of permutations of } 1, 2, \ldots, n^3 \text{ is } n! = n \times (n-1) \times (n-2) \ldots \times 2 \times 1 \]

So then

\[ \text{AVG}_{f_0}(n) = \left\lfloor \frac{n!}{\text{runtime of input}} \right\rfloor \]

\[ n! < \# \text{ of inputs of size } n \]

3 10 100 45 1 67 ...

For input, there are 6 iterations in total, each iteration is 1 step.
Key step:

\[ S_n = \sum_{i=0}^{n-1} (i+1 \text{ steps}) \times \left[ \text{# of permutations that have 1 at index } i \right] \]

Where the 1 is (index)

\[ = \sum_{i=0}^{n-1} (i+1)(n-1)! \]

So in total:

\[ \text{AVG } f_n = \sum_{i=0}^{n-1} \frac{(i+1)(n-1)!}{n!} \in \Theta(n) \]