Palindrome: Algorithm 1

Overview

To determine whether a string is a palindrome, the first algorithm we explored was:

1. Reverse the string.
2. Compare the reversed string to the original string.

For example, the reverse of the string `"noon"` is `"noon"`. Since the reversed string is the same as the original string, `"noon"` is a palindrome.

For the string `"dented"`, the reverse of the string `"dented"` is `"detned"`. Since the reversed string is *not* the same as the original string, `"dented"` is *not* a palindrome.

Code

```
def reverse(s):
    """ (str) -> str

    Return s reversed.

    >>> reverse('hello')
    'olleh'
    """

    s_reversed = ''
    for ch in s:
        s_reversed = ch + s_reversed
    return s_reversed


def is_palindrome_v1(s):
    """ (str) -> bool

    Return True if and only if s is a palindrome.

    >>> is_palindrome_v1('noon')
    True
    >>> is_palindrome_v1('radar')
    True
    >>> is_palindrome_v1('kayaks')
    False
    """

    s_reversed = reverse(s)
    return s == s_reversed
```

Jennifer Campbell • Paul Gries
University of Toronto