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):
    ""\overline{} (str) -> \overline{b}ool
    Return True if and only if s is a palindrome.
    >>> is_palindrome_v1('noon')
    True
    >>> is palindrome v1('radar')
    >>> is palindrome v1('kayaks')
    False
    s reversed = reverse(s)
    return s == s reversed
```

Jennifer Campbell • Paul Gries University of Toronto