University of Toronto - Computer Science - Image

# Palindrome: Approaching the Problem

## Introduction

A palindrome is a word that is read the same front-to-back or back-to-front. For instance, R A C E C A R reversed is also R A C E C A R. However, BLUE reversed is EULB, and thus not a palindrome.

## Algorithms

An *algorithm* is a sequence of steps that accomplish a task. We present three algorithms for detecting palindromes:

### Algorithm 1: Reverse the string. 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.

### Algorithm 2: Split the string into two halves. Reverse the second half. Compare the first half to the reversed second half.

- For example, the first half of the string `"noon"` is `"no"` and the second half is `"on"`. The reverse of the second half is `"no"`. Since the first half is equal to the reverse of the second half, `"noon"` is a palindrome.
- For a string with an odd length, let's consider `"racecar"`. When splitting the string into two halves, we omit the middle character, `"e"`. The first half of the string is `"rac"` and the second half of the string is `"car"`. The reverse of the second half is `"rac"`. Since the first half is equal to the reverse of the second half, `"racecar"` is a palindrome.

### Algorithm 3: Compare the first character to the last, the second to the second last, and so on. Stop when the middle of the string is reached.

- For example, for string `"noon"`, we compare the first character (`"n"`) to the last character (`"n"`), and the second character (`"o"`) to the second last (`"o"`). Since both pairs of characters that were compared are equal, `"noon"` is a palindrome.
- For a string with an odd length, let's consider `"racecar"`. We compare the first character (`"r"`) to the last character (`"r"`), the second character (`"a"`) to the second last (`"a"`), and the third character (`"c"`) to the third last character (`"c"`). The middle character, `"e"` does not need to be compared with anything. Since all pairs of characters that were compared are equal, `"racecar"` is a palindrome.

In upcoming lectures, we will implement all three algorithms. Following the Recipe for Designing Functions, we implemented the function header:

1. Examples
```
>>> is_palindrome('noon')
True
>>> is_palindrome('racecar')
True
>>> is_palindrome('dented')
False
```
2. Type Contract
```
(str) --> boolean
```

3. Header
   `def is_palindrome(s):`
4. Description
   Return `True` if and only if `s` is a palindrome
5. Body
   This will be discussed in future lectures.
6. Test
   This will be discussed in future lectures.

---

Jennifer Campbell • Paul Gries
University of Toronto

---