Team Test 2 on Thu

see website 'Test Info' page
for room + coverage details

---

Last time: palindrome: racecar, hannah,
ai bohphobia (fear of palindromes!)

String prefix: 'sum' 'summation'
'mis' 'mispelt'

Problem: Given a nonempty string S, return the length of the longest prefix that is a palindrome.

Here is an algorithm that solves the problem
def palindrome_prefix(s: str) -> int:
    """Return the length of the longest prefix of a nonempty string $s$
    that is a palindrome.
    """

    >>> palindrome_prefix('attack')
    4

    n = len(s)
    for prefix_length in range(n, 0, -1):  # goes from $n$ down to 1

        # Check whether $s[0:prefix_length]$ is a palindrome
        is_palindrome = True  # assume it is until know otherwise
        i = 0
        while is_palindrome and i < prefix_length:
            if s[i] != s[prefix_length - 1 - i]:
                is_palindrome = False
                i = i + 1

        # if current prefix is a palindrome, return its length
        if is_palindrome:
            return prefix_length

Now: describe $WC_{pp}(n)$

$\max \{ \text{runtime of } pp \text{ on } s \mid \text{len}(s) = n \}$.
Our runtime plot might look like:
$\text{WC}_{\text{pp}}(n) \in O(g(n))$ means

$\exists c_0, n_0 \in \mathbb{R}^+, \forall n \in \mathbb{N}, \ n \geq n_0 \implies$ all below $g(n)$

$\forall$ inputs $s$ with $\text{len}(s) = n$,

runtime $\text{pp}(s) \leq c_0 \cdot g(n)$

$\text{WC}_{\text{pp}}(n) \in \Omega(g'(n))$ means

$\exists c, n \in \mathbb{R}^+, \forall \text{len}(s) = n$,

$c \cdot g'(n) \leq \text{runtime}_{\text{pp}}(s)$.

A lower bound on WC

$g'(n)$ is a lower bound on $\max \{\text{runtime} \mid \text{len}(s) = n\}$

last time: $\text{WC}_{\text{pp}}(n) \in O(n^2)$

stopped $\text{WC}_{\text{pp}}(n) \in \Omega(\_\_\_)$

How? $\Rightarrow$ would like $n^2$ to match $O(\_\_\_)$ bound $\Rightarrow \text{WC}_{\text{pp}}(n) \in \Theta(n^2)$
- Need to construct an example input for each $n$, for which the runtime varies like $n^2$. (an input family)

1. Try $S_n = 'aa ... a'$

   **Description:** $orall n \in \mathbb{N}$,
   
   $S_n$ is a string
   
   $\land \forall i \in \text{range}(n), S[i] = 'a'$

   - The outer loop runs once only
   - The inner loop runs $n$ times
   - Returns $n = \text{len}(S)$, runtime ~ $n$

2. Try $S_n = 'abbb ... b'$

   $orall n \in \mathbb{N}$, $S_n$ is a string
   
   $\land S_n[0] = 'a'$
   
   $\land \forall i \in \text{range}(1, n), S[i] = 'b'$

   - The outer loop runs $n$ times
   - Each inner loop runs 1 time
   - Return 1, runtime ~ $n$ basic operations
What property do we need from $S_n$ to get $n^2$ runtime?

- Need both loops to run a number of times that depends on $n$.

  - For inner loop to keep running
    - Need $S$ to look like a palindrome at ends
    $$S_n = 'aaa ... ... aaa'$$

  - For the outer loop to keep running
    - $S_n$ can't be a palindrome, and location of "failure" point must depend on $n$
      - Put 'b' near middle
    $$S_n = 'aa ... aba ... aa'$$

Update: I originally used $[n/2]$ but should have used $\lceil n/2 \rceil$. Why? because $\lfloor n/2 \rfloor$ puts the 'b' in exact middle for odd $n$, giving a palindrome!
description: binpat family

\[ \forall n \in \mathbb{N}^+, \text{len}(S_n) = n, \ S_n \text{ is a string}, \ S_n \left[ \left\lfloor \frac{n}{2} \right\rfloor \right] = 'b' \]

\[ \forall i \in \text{range} \left( \left\lfloor \frac{n}{2} \right\rfloor, n \right), S_n[i] = 'a' \]

\[ \forall i \in \text{range} \left( \left\lfloor \frac{n}{2} \right\rfloor, n \right), S_n[i] = 'a' \]

aside: \[ \left\lfloor \frac{n}{2} \right\rfloor \]

\begin{align*}
\text{n=4} & : \text{aa ba} \\
\text{n=5} & : \text{aa abaa}
\end{align*}

\[ \left\lfloor \frac{n}{2} \right\rfloor \]

\[ \text{n=10} : \text{aa aaabaaaaaa} \]

\[ \text{n=11} : \text{aaaaa a b aaaa} \]

# iterations is roughly - ignore L \[ \frac{n}{2} \]

\[ \frac{n}{2} + \left( \frac{n}{2} - 1 \right) + \left( \frac{n}{2} - 2 \right) + \ldots + 1 \]

inner loop

prefix is palindrome

running time is at least

\[ = \left( \frac{n}{2} \right) \left( \frac{n}{2} + 1 \right) + \frac{n}{2} \]
\[
= \frac{n^2}{8} + \frac{n}{4} + \frac{n}{2}
\geq \frac{n^2}{8}
\therefore \quad \omega_{p,p}(n) \in SU(n^2)
\therefore \quad \omega_{p,p}(n) \in \Theta(n^2)
\]

What about \( BC_{p,p}(n) = \min \{ \text{runtime}_{p,p} \on s | \text{len}(s) = n \} \)?

Want \( BC_{p,p}(n) \in \Theta(\text{some function}) \)

Find \( BC_{p,p} \in O(h'(n)) \)
and \( BC_{p,p} \in \Omega(h(n)) \)
Our runtime plot might look like:

\[ g(n) \quad g'(n) \]

\[ h(n) \quad h'(n) \]

The BC property means:

\[ \exists c_1, n_2 \in \mathbb{R}^+, \forall n \in \mathbb{N}, \ n \geq n_2 \Rightarrow \]

\[ \forall \text{ inputs } s \text{ with } \text{len}(s) = n, \ c_2 \cdot h(n) \leq \text{runtime}_{pp}(s) \]
BC \( p.p(n) \in O(h(n)) \) means

\[ \exists C_3, n_3 \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \geq n_3 \Rightarrow \exists \text{ inputs with } \text{len}(s) = n, \text{ runtime } p.p \leq C_3 h(n) \]

at least one.
below \( h(n) \)

* note: quantifiers switched from WC

for palindrome prefix:

consider input \( S_n \)

\[ \forall n \in \mathbb{N}, \text{len}(s_n) = n \]
\[ \forall i \in \text{range}(n), S[i] = 'a' \]

The # basic operations is

\[ 1 + 1 + n + 4 \uparrow \]
\[ n = \text{len}(s) \]
\[ \text{inner loop } \leq 2n \text{ for } n \geq 2 \]

\[ \rightarrow BC_{p.p(n)} \in O(n) \]

would give \( BC_{p.p(n)} \in \Theta(n) \)

To show \( BC_{p.p}(n) \in \Omega(n) \)
need to show at least \( C \cdot n \) basic operations performed.

Let \( n \in \mathbb{N}^+ \), and

Let \( S_n \) be an arbitrary string of length \( n \),

and let \( R = \text{palindrome-prefix}(s) \).

Then \( 1 \leq k \leq n \).

The the outer loop executes

\( n - R \) times without finding palindrome

in each case the inner loop executes

at least one

Then (on the final outer loop execution)

the inner loop runs \( R \) times

the total \# of iterations is at least

\[ (n-k) \cdot 1 + R \]

\[ = n \]

\[ \geq 1 \cdot n \]
\[ \therefore \quad BC_{p-p}(n) \in \Omega(n) \]

All together \( BC_{p-p}(n) \in \Theta(n) \).

\[ \Box \]

phew!