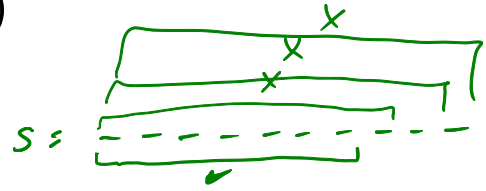
- PS4: complete version is now on Quercus
- TT3... more to follow — but not right now
- TT2: last TA is done, marks out this week!

Last time... is_palindrome(s)

$WC(n) \in \Theta(n)$



```
def pal_prefix(s: str) -> int:
    '''Return length of a longest prefix of s that
    is a palindrome.'''
    for p in range(len(s), 0, -1):  # p = n, n-1, ..., 1
        if is_palindrome(s[0:p]):
            return p
```

(len(s))

$WC_{pal\_prefix}(n) = ?$

- **Upper bound:** Goal: show $RT(s) \leq \ ?$ for all inputs s of size n

Approach:
- overestimate, ignore early termination
- simplify as we go
- danger: we could overcount; check with lower bound, do it more precisely if necessary

- loop body takes time $\leq \not{c} \cdot p \leq \not{c} \cdot n$
- loop iterates $\leq n$
- total is $\leq \not{c} \cdot n \cdot n = \not{c} \cdot n^2$

$\Rightarrow$ $WC(n) \in O(n^2)$

comes from $WC_{is\_pal}(p) \in \Theta(p)$

simplify to just "p" is okay!

- Lower bound:

  Goal: Show $RT(s) \geq$?

  for at least one input s of size n

  → input family: concrete input for each size.

  Approach: – find input family

  – underestimate, simplify as we go

  – danger: leave too much out — check against upper bound, redo as necessary, with more precision.

ROUGH WORK

Trick: • inputs that make pal_prefix iterate

many times are ones where many prefixes of s are not palindromes, e.g.,

s = abcd....
(all letters are different!)

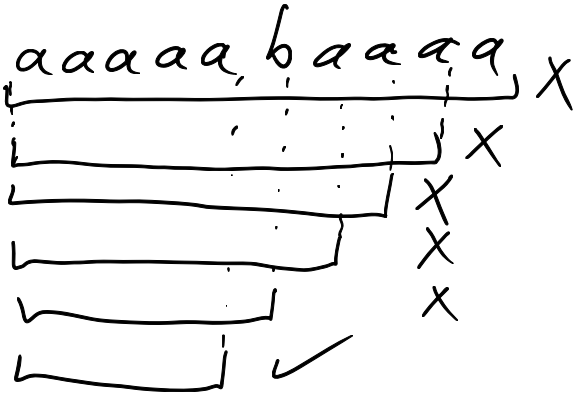
but: is_palindrome runs fast (constant) for such strings...

- inputs that make is_pal take a long time (like s = aaa...a) iterate only once in pal_prefix...

Insight:

let $s = \underbrace{aa\ldots a}_{\lceil \frac{n}{2} \rceil} b \underbrace{aa\ldots a}_{\lfloor \frac{n}{2} \rfloor - 1}$

e.g. n=10: s = aaaaabaaaa



is_pal takes time	$n/2$
is_pal takes time	$n/2 - 1$
	$n/2 - 2$
	$n/2 - 3$
	$n/2 - 4$
	$n/2$
	?

Total time

$$= \lceil \tfrac{n}{2} \rceil + \sum_{p=\lceil \frac{n}{2} \rceil + 1}^{n} \left(p - \lceil \tfrac{n}{2} \rceil \right)$$

(last call to is_pal on first $\lceil \frac{n}{2} \rceil$ characters)

$$\hookrightarrow \sum_{j=1}^{n-\lceil \frac{n}{2} \rceil} j \qquad \left(j = p - \lceil \tfrac{n}{2} \rceil \right)$$

$$n - \lceil \tfrac{n}{2} \rceil = \lfloor \tfrac{n}{2} \rfloor$$

$$\hookrightarrow = \frac{\lfloor \frac{n}{2} \rfloor \left(\lfloor \frac{n}{2} \rfloor + 1 \right)}{2}$$

$\geq \frac{n^2}{8} \implies \underline{WC(n) \in \Omega(n^2)}$

Last example

```
def twisty(n):
    while n > 1:
        if n%2 == 0:
            n = n//2
        else:
            n = 2*n - 2
```

$\Theta(\#\text{iterations})$

Insight: consider multiple iterations...

- n → even → $\frac{n}{2}$ → even → $\frac{n}{4} = n/4$
- n → even → $\frac{n}{2}$ → odd → $2(\frac{n}{2}) - 2 = n-2$
- n → odd → $2n-2$ → even → $\frac{2n-2}{2} = n-1$

} upper bound