Harder WC example

Define palindrome: string s where
\[ s \equiv \text{reverse}(s) \]

Prefix (of a string s):
a substring of s starting at the first character

\[ s[0:k] \]

\[ s = \text{'david'} \]
\[ d \quad \text{dav} \quad \text{david} \]
\[ \text{da} \quad \text{dav} \quad \text{david} \]

def palindrome_prefix(s):
    for k = len(s), len(s) - 1, ..., 2, 1:
        is_pal = True
        # check if s[0:k] is a palindrome
        for j = 0...k-1:
            if s[j] \neq s[k-1-j]:
                is_pal = False
                break (out of inner loop)
        if is_pal:
            return k

Ex:
\[ \text{pal-pre('abba david')} = 4 \]

Goal: find bounds on WC_{pal-pre}(n), where
\[ n \text{ is the length of the input string.} \]
Easy part: Prove that $W_{C,p}(n) \in O(n^2)$.

Key idea: Ignore the early breaks/returns, and get an upper bound on the number of iterations of each loop.

Exercise: "Each loop runs $\leq n$ iterations" (go from there)

Hard part: Prove that $W_{C,p}(n) \in \Omega(n^2)$.

Need to find an input family (one input for each $n \in \mathbb{N}$) whose runtime is $\geq cn^2$.

Bad input families $\Rightarrow$ they run too fast

1. $s$ is a palindrome:
   
   Let $n \in \mathbb{N}$. $s = \overline{a \ldots a}$ ($n$ times)
   
   $\Rightarrow W_{C,p}(n)$
1. inner loop runs \( n \) times, but
\[ x \rightarrow \text{outer loop stops at first iteration!} \]

2. Let \( n \in \mathbb{N} \). Pick \( s = \overline{a \ldots b} \)
\[ \text{n-1 times} \]

\[ x \rightarrow \text{inner loop runs 1 time (per outer loop iteration)} \]

\[ \rightarrow \text{outer loop runs } n \text{ times} \]

"Good" input family

Let \( n \in \mathbb{N} \). Pick \( s = \overline{a \ldots a\overline{b\ldots a}} \)
\[ \left\lfloor \frac{n}{2} \right\rfloor + 1 \]

**Example**

**Best-case**

\[ \text{WC}(n) \]

\[ \text{BC}(n) \]

\[ \text{BC}(n) \leq O(g(n)) \rightarrow \text{existential (3 an input family w/ runtime } \]
Runtime Recap

1. "Simple": runtime depends only on input size

2. "Harder": runtime varies, even when given inputs of the
given inputs of the same size

\[ WC(n) = \max \{ \text{runtime on input size } n \} \]

\[ BC(n) = \min \{ \text{runtime on size } n \} \]

describe using \( O, \Omega, \Theta \).

because we don't know exactly what they are!

Sometimes worst-case is too pessimistic

\[ WC(n) \in \Theta(2^n) \]
For a program $P$, we define the average-case runtime of $P$ to be the function: $\text{Avg}_p(n) : \mathbb{N} \to \mathbb{R}^\geq 0$ as

$$\text{Avg}_p(n) = \text{average \{ runtime of } P(x) \text{ of } x \text{ has size } n \}$$

\( \text{total of all runtimes} \) \( \# \text{ of inputs} \)

\( \rightarrow \text{Avg}_p(n) \) depends on how we choose our inputs!

**Example**

```python
def find_one(L):
    for i in range(L):
        if L[i] == 1:
            return i
```

1 iteration

\[ 1 \ ? \ ? \ ? \ ? \ ? \]

5 starts
if \( L[i] = 1 \):
    return

Find a bound on \( \text{Aug}_{\text{f}o}(n) \), given the set of inputs:

For each \( n \in \mathbb{N} \), the \underline{permutations} of \( \{1, 2, \ldots, n\} \):

\begin{align*}
\text{n=0} & \quad [\ ] \\
\text{n=1} & \quad [1] \\
\text{n=2} & \quad [1, 2], [2, 1] \\
\text{n=3} & \quad [1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]
\end{align*}

Fact: there are \( n! \) inputs of length \( n \),

\[ n \times (n-1) \times (n-2) \times \cdots \times 3 \times 2 \times 1 \]
$\text{Avg}(n) = \frac{\text{Sum of runtimes for size } n \text{ inputs}}{\# \text{ of size } n \text{ inputs}}$

$= \frac{\text{sum} \ldots}{n!}$

Goal: calculate $S_n$.

$\Rightarrow$ We need to group the inputs by their runtime.
( Need to understand the code! )

The only thing that matters is the position of the 1 in the input.
So group by the position at the 1.

\[ S_n = \text{sum of all runtimes for size n inputs} \]
\[ = (\text{sum of runtimes where 1 is at index 0}) \]
\[ + (\text{sum of runtimes where 1 is at index 1}) \]
\[ + \ldots \]
\[ + (\text{sum of runtimes where 1 is at index } n-1) \]
\[ = (1 \text{ step}) \times \# \text{ of inputs w/ 1 at index0} \]
\[ + (2 \text{ steps}) \times \# \text{ of inputs w/ 1 at index1} \]
\[ + \ldots \]
\[ + (n \text{ steps}) \times \# \text{ of inputs w/ 1 at index}(n-1) \]
\[ \times (n-1)! \text{ of them} \]
\[\begin{align*}
&= (1 \text{ step}) \,(n-1)! \\
&\quad + (2 \text{ steps}) \,(n-1)! \\
&\quad + \ldots \\
&\quad + (n \text{ steps}) \,(n-1)! \\
&= \frac{n \,(n+1)}{2} \,(n-1)! \\
&= \frac{n+1}{2} \,(n!)
\end{align*}\]

\[\begin{align*}
\text{Avg}(n) &= \frac{\sum_{n}^{n}}{n!} \\
&= \frac{n+1}{2} \cdot n! \\
&= \frac{n+1}{2} \cdot n! \\
&= \frac{n+1}{2} \in \Theta(n)
\end{align*}\]