This Week

Midterm 2
Problem set 4 due

Last Problem set out
Fri/Saturday

Today: Another worst-case analysis
Average-case runtime
```python
Def PalZ(s)  # s is a string
    n = len(s)

    For k in range(n, 1, -1)  # k = n, n-1, ..., 1
        IS_PAL = true

        For j in range(k):
            # Check if s[0..j-1] is a palindrome
            if s[j] != s[k-j-1]:
                IS_PAL = FALSE
                BREAK  # Out of inner loop

    If IS_PAL:
        # We've found longest palindrome prefix
        return j
```
Example

With "Break"  Without "Break"

\[ n = 4 \]
\[ a a a a b c d \]

\[ k = n = 7 \]
\[ k = n - 1 = 6 \]
\[ k = n - 2 = 5 \]
\[ k = n - 3 \]
\[ n = 7 \]
\[ n - 1 = 6 \]
\[ n - 2 = 5 \]
\[ n - 3 = 4 \]

\[ n = 7 \]
\[ n = 6 \]
\[ n = 5 \]
\[ n = 4 \]

Successful iteration of outer loop

Unsuccessful iterations of outer loop
Worst Case Runtime

$$W_{\text{Pal2}} (n) \in O(n^2)$$

We want to show

\[ \text{In steps (if size of S is n then alg on S takes) } \leq n^2 \text{ steps} \]

when \( k=n \), inner loop executes at most \( n \) times

more generally

when \( k=i \), inner loop executes at most \( i \) times

Total number of times inner loop executes is

\[ \leq n + (n-1) + (n-2) + \ldots + 1 \leq O(n^2) \]
\begin{align*}
\frac{n(n+1)}{2} & \leq \frac{n^2}{4} + \frac{n}{2} \leq \frac{n^2}{4} + \frac{n^2}{2} \\
& \leq \frac{3n^2}{4} \\
\text{let } \eta_0 = 1, \quad c_0 = \frac{3}{4} \\
\text{gives us } O(n^2)
\end{align*}
Now we want to show

\[ \text{WC}_{\text{PAL2}}(n) \in \Theta(n^2) \]

\[ \forall n \in \mathbb{N} \ (\text{size of } S \text{ is } n \text{ and runtime of PAL2 on } S \text{ is } \Theta(n^2)) \geq c \cdot n^2 \]

Last class (without "break" stmt)

We used this string:

\[ \text{a b b b ... } b \]

Now this string isn't hard enough — alg. runs in time \( \Theta(n) \)
Let $S$ contain all a's except for one b that is one position to the right of the middle. b occurs in position $\Gamma_{n/2}$

```
| a | a | a | a | a | a | a | b | a | a | a |
```

$n = 10$

```
| a | a | a | a | a | a | a | b | a | a | a |
```

$n = 9$

Run of algorithm on aaaaabaaaaaa (n=10)

- $K = n = 10$: inner loop executes $K - \lceil n/2 \rceil = 10 - 5 = 5$ times
- $K = n-1 = 9$: inner loop executes $K - \lceil n/2 \rceil = 9 - 5 = 4$ times
- $K = \lceil n/2 \rceil + 1$: inner loop executes $K - \lceil n/2 \rceil = 1$ time
Let's analyze runtime on this S

(1.) Unsuccessful iterations of the outer loop

\[ k = n \quad k - \left\lfloor \frac{k}{2} \right\rfloor = n - \left\lfloor \frac{n}{2} \right\rfloor \quad \text{inner loop executions} \]

\[ k = n - 1 \quad k - \left\lfloor \frac{k}{2} \right\rfloor = n - 1 - \left\lfloor \frac{n}{2} \right\rfloor \quad \text{inner loop executions} \]

as long as \( k \geq \left\lfloor \frac{n}{2} \right\rfloor + 1 \)

(2.) when \( k = \left\lfloor \frac{n}{2} \right\rfloor \)

successful iteration

inner loop will run for \( \left\lfloor \frac{n}{2} \right\rfloor \) steps

so (1) alone gives

\[ n - \left\lfloor \frac{n}{2} \right\rfloor + (n-1) - \left\lfloor \frac{n}{2} \right\rfloor + \ldots + \left\lfloor \frac{n}{2} \right\rfloor + 1 - \left\lfloor \frac{n}{2} \right\rfloor \leq \left( \frac{1}{2} - 1 \right) + \left( \frac{3}{2} - 2 \right) + \ldots + 1 \]
\[ \sum_{i=1}^{m} \frac{m!}{i!} = \frac{(m)(m+1)}{2} \]

\[ K = n \quad \sim \quad \frac{n}{2} \]

\[ K = n-1 \quad \sim \quad \frac{n}{2} - 1 \]

\[ K > \frac{n}{2} + 1 \quad \sim \quad K - \frac{n}{2} \]

\[ = \ \frac{n}{2} + \frac{n}{2} - 1 + \frac{n}{2} - 2 + \ldots + 1 \]

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

\[ = \ \frac{n^2}{4} + \frac{n}{2} \]

\[ = \ \frac{n^2}{8} + \frac{n}{4} \]
\[
\frac{n^2}{8} + \frac{n}{4} \geq \frac{n^2}{8} = \mathcal{O}(n^2)
\]

So far we know

\[
WC_{PAL2}(n) = \Theta(n^2)
\]
Best case complexity

Before (without "break" statement)
we shared best-case complexity was $\Theta(n)$

1. $BC_{PAL}(n) \in O(n)$
2. $BC_{PAL}(n) \in \Omega(n)$

1. $\forall n \exists s$ (size of $s$ is $n$ and runtime on $s$ is $\leq (n)$)
2. $\forall n \forall s$ (if $s$ has size $n$ then runtime $\geq n$)
New analysis for best case complexity for PAL2

$$\forall n \forall s \ (\text{if } s \text{ has length } n \text{ then runtime } \geq n)$$

2 cases

(a) If $s$ is a palindrome of length $n$ then runtime $\geq n$

(outer loop runs once; inner loop $n$ times)

(b) If $s$ is not a palindrome
(b) cont'd

Let's say that $S_{i-5}$ is the longest prefix that is a palindrome.

ex. $\overline{aaaaaabcdef} \\
1 2 3 4 5$

$i = 5$

When $K > i$ outer loop executes and inner one executes $\geq 1$ steps.

When $K = i$ outer loop executes $\geq 1$ times.

$\text{Total steps} = n - i + i = n$

so total steps $\geq n - i + i = n$. 
Let $s$ be any length $n$ palindrome. Then alg on $s$ takes $n$ steps.
Average Case Runtime Analysis

Worst Case $f(n)$:  $\max_{x \text{ a length} \leq n} (\text{Runtime of } f \text{ on } x)$

Best Case $f(n)$:  $\min_{x \text{ a length} \leq n} (\text{Runtime of } f \text{ on } x)$

Average $f(n)$:  supposed to capture the runtime of $f$ on a "typical" $x$
Let \( T_n = \{ x \mid x \text{ has size } n \} \)

Consider all inputs to \( f \) of size \( n \)

\[
\text{Average}_{f}(n) = \frac{\sum \text{Runtime of } f \text{ on } x}{|T_n|}
\]
Example

Def Find1(L)
    n = len(L)
    For i = 1, 2, ..., n
        If L[i] = 1
            halt and output i

L is an array of length n
each number in 1..n appears exactly once
in the list L

WC_{Find1}(n) \leq \Theta(n)

BC_{Find1}(n) = \Theta(1)

n = 8
ex 7 2 1 3 4 5 8 6
Let's show that \( \text{average } f(n) = O(n) \)

What is \( T_n \)? How big is \( T_n \)?

\( T_n = \text{set of all inputs to } \text{Finds } q \text{ length } n \)

\[ |T_n| = n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 1 \]

There are \( n \) choices for \( L[1] \), \( (n-1) \) choices for \( L[2] \), and so on.

So altogether \( n \cdot (n-1) \cdot \ldots \cdot 2 = n! \) many different orderings/permutations are possible.
\[ \text{Average } \text{Finds}_3(n) = \sum_{L \in T_n} \frac{\text{Runtime of } \text{Finds on } L}{|T_n|} = \frac{\sum \text{runtime on } L}{n!} \]

Let's let \( T_n^i = \{ L \in T_n \mid L[i] = 1 \} \)

\[ T_n = T_n^1 \cup T_n^2 \cup \ldots \cup T_n^n \]

We require these sets \( T_n^i \) to be disjoint so every \( L \in T_n \) is in exactly one \( T_n^i \).
What is the runtime for any \( L \in T_n^i \)? It is:

\[
\text{Average}_{\text{time}}(n) = \frac{1}{n!} \left( \sum_{i=1}^{n} |T_n^i| \cdot \frac{1}{i} + |T_n^2| \cdot \frac{1}{2} + |T_n^3| \cdot \frac{1}{3} + \ldots + |T_n^n| \cdot \frac{1}{n} \right)
\]

So we just have to compute \( |T_n^i| \) for every \( i \):

\[
|T_n^1| = (n-1)!
\]
For every \( i \): \( |T_n^i| = (n-1)! \)

So:

\[
\text{Average}_{\text{find1}}(n) = \frac{(n-1)! \cdot 1 + (n-1)! \cdot 2 + \ldots + (n-1)! \cdot n}{n!} = \frac{(n-1)!}{n!} \left( \sum_{i=1}^{n} i \right) = \frac{1}{n} \cdot \frac{n(n+1)}{2} = \frac{n+1}{2}
\]