CSC165 fall 2017

Danny Heap
csc16517f@cs.toronto.edu
BA4270 (behind elevators)
Web page:
http://www.teach.cs.toronto.edu/~heap/165/F17/
416-978-5899

Using Course notes: more Induction
Outline
compare...

def is_prime(n):
    if n < 2:
        return False
    else:
        for d in range(2, n):
            if n % d == 0:
                return False
        return True

def has_even(number_list):
    for number in number_list:
        if number % 2 == 0:
            return True
    return False

even though different
for different, exactly 1 input
for each size n

n = len(number_list)
Runtime depends
on list contents —
where 1st even # is.
**definitions**

Set of inputs of size \( n \)

\[ I_{f,n} = \{ i \mid i \text{ is an input to } f \land |i| = n \} \]

- \( RT_f(x) = \) number of basic "steps" in executing \( f(x) \)

  *steps independent of \( |x| \)*

- \( WC_f(n) = \max\{RT_f(x) \mid x \in I_{f,n}\} \rightarrow \text{longest-running instance} \)
upper bounds, lower bounds...

U(n) is an upper bound means
\[ \forall n \in \mathbb{N}, \forall x \in \mathcal{I}_{f,n}, RT_f(x) \leq U(n) \]

L(n) is a lower bound means
\[ \forall n \in \mathbb{N}, \exists x \in \mathcal{I}_{f,n}, RT_f(x) \geq L(n) \]

why the asymmetry of U and L?
\( WC_{\text{has\_even}} \in O(n) \)

Loop executes \( \leq n \) times

"return False" executes \( \leq 1 \) time

Proof: Let \( n+1 \) be an upper bound on \( WC_{\text{has\_even}} \)

Let \( n \in \mathbb{N} \). Let \( L \) be an arbitrary list of \( \text{ints of size } n \), i.e. \( \text{len}(L) = n \). Then \( \text{has\_even}(L) \) costs

- \( \leq n \) "steps" for loop
- \( \leq 1 \) "step" for return False

\( \leq n+1 \) steps, hence \( \in O(n) \)
$WC_{\text{has\_even}} \in \Omega(n)$

- dream up $f : I_{\text{has\_even}} \leftarrow n$ such that each element of $f$ costs a lot.

Here $|L| = n$, $L[i] = 5 \forall i \in \text{range}(n)$.

**Proof** $\geq n+1$ is a lower bound on $WC_{\text{has\_even}}(n)$.

Let $n \in \mathbb{N}$. Let $L$ be list of $n-5$s.

Then an instance of $\text{has\_even}(L)$ costs:

- $n$ iterations of loop $\rightarrow n$ steps since "return" within loop never executes
- 1 step to return "False"

So, $n+1$ steps $\geq L$
palindromes

examples: “racecar rotor pap...” every string starts with a palindrome, so find the longest palindrome prefix...

def palindrome_prefix(s):
    n = len(s)
    for prefix_length in range(n, 0, -1):  # count down from n
        is_palindrome = True
        for i in range(prefix_length):
            if s[i] != s[prefix_length - 1 - i]:
                is_palindrome = False
                break
        if is_palindrome:
            return prefix_length

WC_{pp} (n) \quad U(n) - easy to overestimate & show O(n^2)
average...

\[ I_{f,n} = \{ i \mid i \text{ is an input to } f \land |i| = n \} \]
\[ \mathcal{T}_{f,n} = \{ t \mid \exists x \in I_{f,n}, t = RT_f(x) \} \]

def has_even(number_list):
    for number in number_list:
        if number % 2 == 0:
            return True
    return False