CSC165 fall 2017

worst/best/average

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

notes

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
```

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

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

n = len(number_list)

runtime depends on list contents — where 1st even # is.

definitions

Set of inputs of size n

- $\mathcal{I}_{f,n} = \{i \mid i \text{ is an input to } f \wedge |i| = n\}$

- $RT_{f(x)} =$ number of basic "steps" in executing $f(x)$

  steps independent of $|x|$

- $WC_f(n) = \max\{RT_{f(x)} \mid x \in \mathcal{I}_{f,n}\} \rightarrow$ longest-running instance

upper bounds, lower bounds...

on worst-case

- $U(n)$ is an upper bound means
  $\forall n \in \mathbb{N}, \forall x \in \mathcal{I}_{f,n}, RT_{f(x)} \leq U(n)$

on worst case

- $L(n)$ 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?

max

$L \leq \max$

$\Leftrightarrow \exists x \geq L$

L

$U \geq \max$

$\Leftrightarrow \forall x \in S, x \leq U$

$WC_{\text{has\_even}} \in O(n)$

loop executes $\leq n$ times

"return False" executes ≤ 1 time

$WC_{\text{has\_even}} \in O(n)$

Proof $\underline{U(n)} = n+1$ is an upper bound on $WC_{\text{has\_even}}$

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

- $\leq n$ "steps" for loop
- ≤ 1 "step" for return False

$\leq n+1$ steps, hence $\in O(n)$ ∎

$WC_{\text{has\_even}} \in \Omega(n)$

- dream up $F \subseteq I_{\text{has\_even}, n}$, such that each element of F costs a lot.

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

Proof $\underset{L=}{}$ $n+1$ is a lower bound on $WC_{\text{has\_even}}(n)$

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

Then an instance of 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 $\geq n$

palindromes

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

$WC_{pp}(n)$

$U(n)$ —

easy to overestimate

& show $O(n^2)$

```
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
```

average...

$$\mathcal{I}_{f,n} = \{i \mid i \text{ is an input to } f \wedge |i| = n\}$$
$$\mathcal{T}_{f,n} = \{t \mid \exists x \in \mathcal{I}_{f,n}, t = RT_f(x)\}$$

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