PS2: graded + returned...

PS3 - due tomorrow - questions

PS4: soon, ... soon

T2: less soon

FE ...

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

upper bounds, lower bounds...

Saves work: ① don't have to find + prove actual worst-case input
② don't have to calculate exact number of steps

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

just need a bad input, not nec. worst.
- no need to calculate exact steps, e.g. can focus on costly part of code

- $L(n)$ 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?

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
```

— step

— step

$\leq n$ times

step

$\leq n$ times

$+ 1$

$1 + 2n + n^2$

$\in O(n^2$

palindromes

$S = \text{"}a \cdots a|ba \cdots a\text{"}$

$s[\lceil n/2 \rceil]$

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

steps: – $0, \ldots, \lceil n/2 \rceil - 1 \rightarrow \lceil n/2 \rceil$

$\rightarrow +1 + 2 + \cdots + \underbrace{n-1-\lceil n/2 \rceil + 1}_{\lfloor n/2 \rfloor}$

$= 1 + 2 + \cdots + \lceil n/2 \rceil = \dfrac{\lceil n/2 \rceil(\lceil n/2 \rceil + 1)}{2} \geq cn^2$ some c

$\in \Omega(n^2)$

```
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...

Probably need to restrict inputs

problem?

assume uniform dist

$$\frac{\sum_{t \in \mathcal{T}_{f,n}} t}{|\mathcal{I}_{f,n}|}$$

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

$$\mathcal{T}_{f,n} = \{t \mid \exists x \in \mathcal{I}_{f,n}, t = RT_f(x)\}$$ - possibly infinite

∴ assume number $\in \mathbb{Z}$

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

restrict to lists from $\{0, 1\}$

$[0, 0] \rightarrow RT = 1$

$[0, 1] \rightarrow 1$

$[1, 0] \rightarrow 2$

$[1, 1] \rightarrow 3$

$$\frac{\sum = 7}{4}$$

average...

lists of length 3 $\in \{0,1\}$

8 (denom)

1 step $\longrightarrow$ 4 4

2 steps $\longrightarrow$ 2 4

3 steps $\longrightarrow$ 1 3

4 steps $\longrightarrow$ 1 4

$\frac{15}{8}$

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

$$\mathcal{T}_{f,n} = \{t \mid \exists x \in \mathcal{I}_{f,n}, t = RT_f(x)\}$$

lists of len 4: $\frac{31}{16}$

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

list length

$2^{n-1} \cdot 1 + 2^{n-2} \cdot 2 + \cdots + 2^{n-n} \cdot n + (n+1)$

$2^n \sum_{i=1}^{n} i \left(\frac{1}{2}\right)^i$

$2^{-i} = \left(\frac{1}{2}\right)^i$

$\Sigma = \sum_{1}^{n} (i-1) r^{i-1}$

$\sum_{i=0}^{n-1} i r^i = \frac{n r^n}{1-r} + \frac{r - r^{n+1}}{(1-r)^2}$

Notes