ps3: posted by tomorrow 10 p.m EST...

- About that Marks hour, ...

- ps1: 11/373 unmarked! → supposed to be 5 p.m. today

CSC165 fall 2017

counting steps...

Danny Heap

csc16517f@cs.toronto.edu

BA4270 (behind elevators)

Web page:

http://www.teach.cs.toronto.edu/~heap/165/F17/

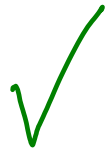
416-978-5899

Using Course notes: more Induction

Outline

notes

big-Theta means...



$$g \in \Theta(f) : \exists c_1, c_2, n_0 \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \geq n_0$$
$$\Rightarrow g(n) \leq c_1 f(n) \wedge g(n) \geq c_2 f(n)$$

products and sums redux

- $f + g$
- af
- $f \cdot g$

back to code...

“steps” and input “size”

counting loops...

```
def f0(n):
    x = n
    print(x * 2)
    return x + 3
```

```
def f1(n):
    for i in range(10):
        print(n)
```

```
def f2(n):
    for i in range(n):
        print(n)
```

$\Theta(n)$

```
def f3(n):
    i = 0
    while i*i < n:
        print(i)
        i = i + 1
```

$s = i$

exit

$s^2 \geq n$

$s \geq \sqrt{n}$

$s = \lceil \sqrt{n} \rceil$

```
def f4(n):
    i = 0
    while i**(1/2) < n:
        print(2*i)
        i = i + 1
```

$s = i$

exit

$\sqrt{s} \geq n$

$s \geq n^2$

$s = n^2$

nested loops

```
def f5(n):
    for i in range(0, n, 2):
        for j in range(n):
            print(i - j)
```

$\Theta(\frac{n^2}{2}) \equiv \Theta(n^2)$

$n \times \lceil \frac{n}{2} \rceil$

eg

for i in range(n):
  for j in range(n):
    x = 5

$n + n + n + \ldots + n$

i goes 0, 2, 4, ... $i = 2s$

$i = 2s \geq n$

$s \geq n/2 \cdot \; s = \lceil \frac{n}{2} \rceil$

```
def f6(n):
    for i in range(n):
        for j in range(i):
            print(i - j)
```

iterates n times

but inner iterates i times

$0 + 1 + \ldots + n-1 = \frac{(n-1)(n)}{2}$

Must use series — $\Theta(n^2)$

$\frac{n^2}{2} - \frac{n}{2} \in O(n^2)$ $c = 1$ — $\frac{n^2}{2} - \frac{n}{2} \in \Omega(n^2)$ $c = \frac{1}{4}$ $n_0 = 2$

composition, combination

```
def f7(n):
    for k in range(n):
        f6(n)
        f2(n)
```

$\Theta(n^2)$

$\Theta(n)$

by ≈ Thm 5.6 $\Theta(n^2)$

n times

$\Theta(n^3)$

$s = k$

exit $s = k \geq n$

$s = n$

I think relevant to ps3

```
def twisty3(n):
    steps = 0
    while n > 0:
        if n % 3 == 0:
            n = n // 3
        elif n % 3 == 1:
            n = 3*n - 3
        else:
            n = 3*n - 6
        steps = steps + 1
        print(n, steps)
    return steps
```

n	$RT_{twisty3}(n)$
0	0
1	1
2	1
3	2
4	4
5	4
6	2
7	4
8	4
⋮	

clumsy is_prime

37-2+1

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

1 step

1 step

n	$RT_{is\_prime}(n)$
35	→ 5 +1?
36	→ 2 +1?
37	→ 37 +1?

RT_{is_prime}

$O(n)$

$\Omega(1)$

Notes