Lecture 12

Tuesday Oct. 17

RT = summation of RT's of all primitive operations

$$= \sum_{i=1}^{N} \boxed{t_i}\,(pm_i)$$

c

$$= C \cdot \sum_{i=1}^{N} pm_i = C \cdot N$$

$$\approx N$$

2 ops.

op1.	$t(op1)$
	$+$
op2.	$t(op2)$

$prog_A$ more efficient than $prog_B$

$\Rightarrow$ # p.o. of A

$\leq$ # p.o. of B

Example: Counting # of Primitive operators.

```
findMax (int[] a, int n) {
  currentMax = a[0];
  for (int i = 1; i < n; ) {
    if (a[i] > currentMax) {
      currentMax = a[i]; }
    i ++ }
  return currentMax; }
```

i	i<n
1	T
2	T
⋮	
n-1	T
n	F

e.g. findMax({2, 3, 4}, 3)

input size (a.length).

$\cancel{7}\cdot n^1 + \cancel{2\cdot n^0}$

$\cancel{3}n - \cancel{2}$

n == 10

i	i<10
1	T
2	T
⋮	
9	T
10	F

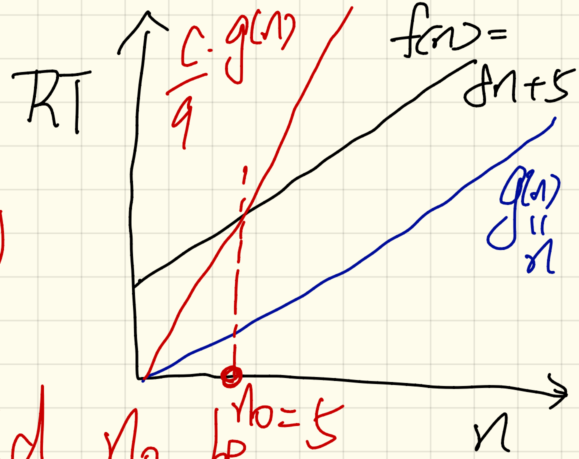
Example: Bounding Function

input size

n	$f(n)$ (code) $8n + 5$	$9n$
1	13	9
2	21	18
3	29	27
4	37	36
5	45	45
6	53	54

...

$$9 \cdot n$$
$$c \quad g(n)$$



What should n_0 be starting from which $f(n) \le c \cdot g(n)$
5
9

before $n_0 = 5$, no upper bound effect!

$f(n) = 8n + 5$ (your code)

$g(n) = n$

Show that $f(n)$ can be bounded by $g(n)$:

choose $c = 9$.

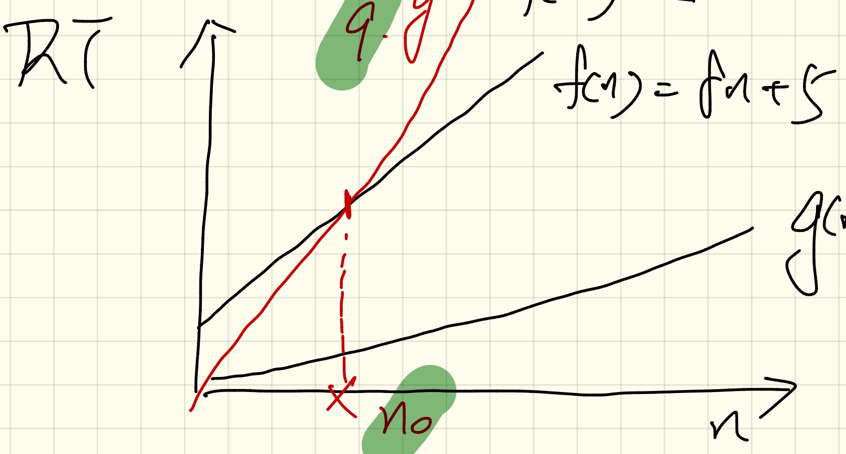
$f(n) = 8n + 5$

Show: $f(n)$ is $O(n)$

$g(n)$

Show that starting from n_0,

we have $f(n) \le c \cdot g(n)$



$9 \cdot g(n)$, $9 \to c$

$$f(n) = a_0 \cdot n^0 + a_1 \cdot n^1 + \cdots + a_d \cdot n^d$$

$$\le a_0 \cdot n^d + a_1 \cdot n^d + \cdots + a_d \cdot n^d$$

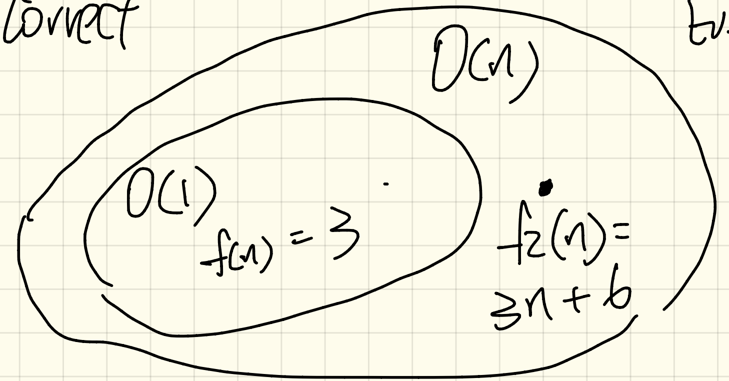
$$\le \underbrace{(|a_0| + |a_1| + \cdots + |a_d|)}_{C} \cdot n^d$$

n is input size

$\Rightarrow n > 0$

$\Rightarrow n^0 < n^1 < n^2 < n^3 \cdots$

3 is $O(1)$ correct
is $O(n)$



Every function that is $O(1)$ is also $O(n)$

Not vice versa.