Section 8.2
Complexity

BIG O

O for "order" (of magnitude)

Let f and g be functions $\mathbb{N} \to \mathbb{R}$.
We say that "f is big O of g" and write

$$f = O(g) \quad \text{or} \quad f \in O(g)$$

if there is a natural number n_0 and a positive real number c such that

$$|f(n)| \le c\,|g(n)|$$

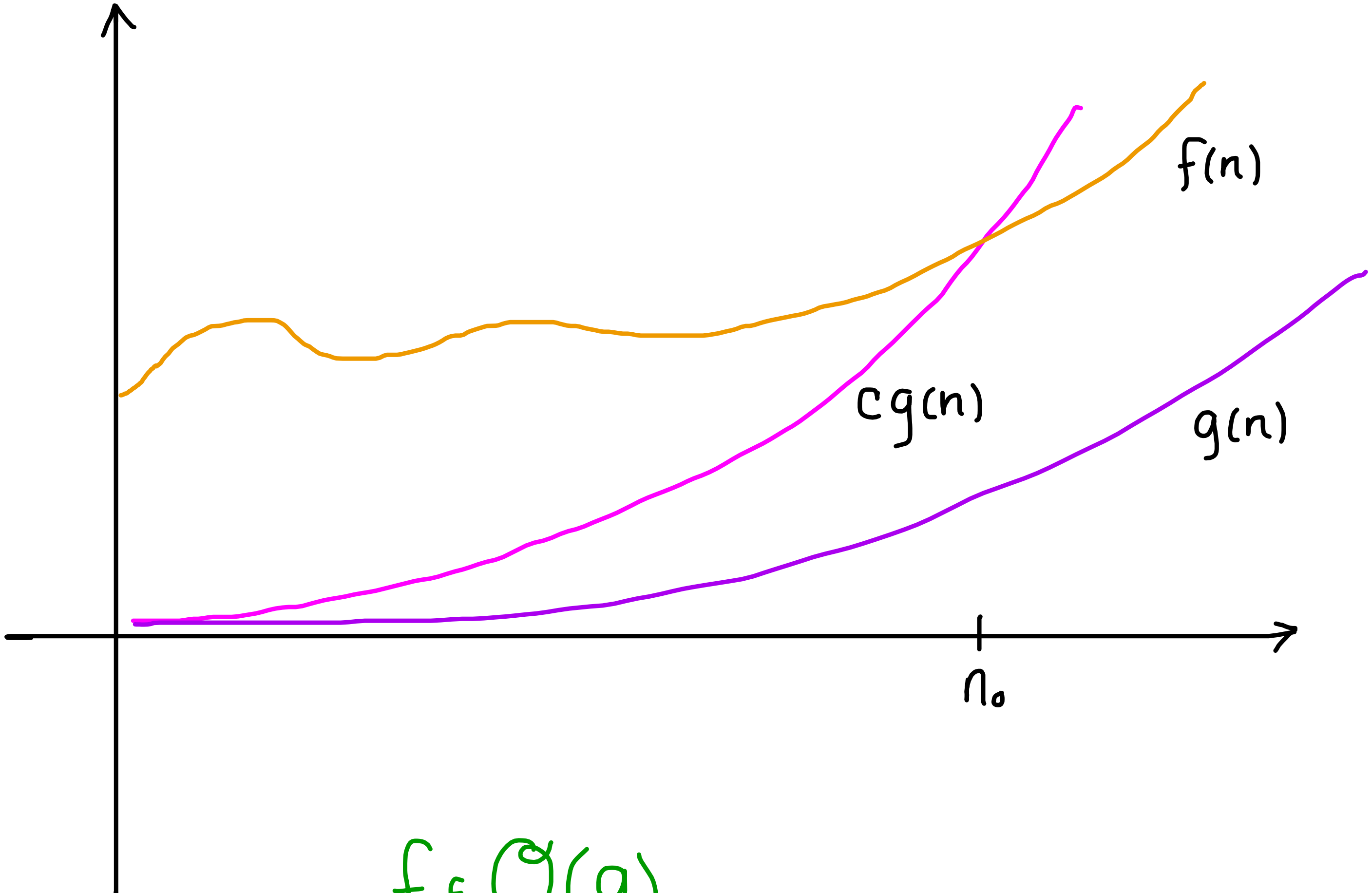
for $n \ge n_0$. "for large n"

Note: If $f, g : \mathbb{N} \to [0, \infty)$ we can drop the absolute values.

Note: There are infinitely many choices for n_0 and c.

Observation: If $f(n) \le g(n)$ for all n, then f is $O(g)$

BIG O



$f \in O(g)$

BIG O

We say that "f is big O of g" and write

$$f = O(g) \quad \text{or} \quad f \in O(g)$$

if there is a natural number n_0 and a positive real number c such that

$$|f(n)| \le c\,|g(n)|$$

for $n \ge n_0$.

First examples: 1 $f(n) = n^2$, $g(n) = 7n^2$

$f \in O(g)$ $\quad c = 1,\ n_0 = 1$

$g \in O(f)$ $\quad c = 7,\ n_0 = 1$

2 $f(n) = 4n+2$, $g(n) = n$

$f \in O(g)$ $\quad c = 5,\ n_0 = 2$

$g \in O(f)$ $\quad c = 1,\ n_0 = 1$

ANOTHER EXAMPLE

Example: $f(n) = n^2$, $g(n) = n^2 + n$

NOT BIG O

How do we show f is <u>not</u> $\mathcal{O}(g)$?

Need to show no c, n_0 work.

Example: $f(n) = n \quad g(n) = \sqrt{n}$

COMPARING FUNCTIONS

Let f and g be functions $\mathbb{N} \to \mathbb{R}$.

We say...	and write...	if...
f has smaller order than g	$f \prec g$	
f has the same order as g	$f \asymp g$	

MORE EXAMPLES

Show that $5n^3 + 12n \asymp n^3$

Show that $n+1 \asymp n$

MORE EXAMPLES

① Compare $n!$ & n^n

② Compare $n!$ & 2^n

COMBINING FUNCTIONS

Theorem: Let f, g be functions $\mathbb{N} \to \mathbb{R}$.

(a) If $f \in \mathcal{O}(F)$, then $f + F \in \mathcal{O}(F)$

(b) If $f \in \mathcal{O}(F)$ and $g \in \mathcal{O}(G)$ then $fg \in \mathcal{O}(FG)$.

Proof:

For example, $(n+1)(5n^3+12n) = 5n^4 + 5n^3 + 12n^2 + 12n$ is $\mathcal{O}(n^4)$ by (b)

What about $19n^{58} + n^{18} - 3n^{10}$?

BIG O VIA LIMITS

THEOREM: Let f, g be functions $\mathbb{N} \to \mathbb{R}$.

(a) If $\lim_{n\to\infty} f(n)/g(n) = 0$, then $f \prec g$

(b) If $\lim_{n\to\infty} f(n)/g(n) = \pm\infty$, then $g \prec f$

(c) If $\lim_{n\to\infty} f(n)/g(n) = L \neq 0$, then $f \asymp g$

PROOF: (a) $\lim f(n)/g(n) = 0$ means: For all $\varepsilon > 0$, there exists n_0 so that $|f(n)/g(n)| < \varepsilon$ when $n \geq n_0$. In other words

$$|f(n)| < \varepsilon |g(n)| \quad , \quad n \geq n_0 \qquad (*)$$

$\rightsquigarrow f \in O(g)$

On the other hand, need $g \notin O(f)$.

$g = O(f)$ means $|g(n)| \leq c|f(n)| \quad n \geq n_0$

i.e. $\frac{1}{c}|g(n)| \leq |f(n)| \quad n \geq n_0$

contradicting $(*)$ ▨

POLYNOMIALS

Theorem: Let $f(n) = a_d n^d + \cdots a_1 n + a_0$ be a degree d polynomial $(a_d \neq 0)$. Then $f(n) \asymp n^d$.

Can prove using either of the last two theorems.

Proof:

MORE COMPARISONS

Theorem: (a) If $k<\ell$, then $n^k < n^\ell$

(b) If $k>1$, then $\log_k n < n$

(c) If $k>0$, then $n^k < 2^k$

Proof:

HIERARCHY

$1 < \log n < n < n^k < k^n < n! < n^n$

const < log < linear < poly < exp < fact < tower

MORE DETAILED HIERARCHY

$$1 < \log n < \sqrt{n} < n/\log n < n < n\log n < n^{3/2}$$

$$< n^2 < n^3 < \cdots$$

$$< 2^n < 3^n < \cdots$$

$$< n!$$

$$< n^n < n^{n^n} < \cdots$$

COMPARING DIFFERENT ORDERS

	10	50	100	300	1000
$5n$	50	250	500	1500	5,000
$n \log n$	33	282	665	2469	9966
n^2	100	2500	10,000	90,000	1,000,000
n^3	1,000	125,000	1 mil	27 mil	1 bil
2^n	10^{24}	16 digits	31 dig.	91 dig.	302 dig.
$n!$	3.6 mil	65 dig.	161 dig.	623 dig.	unimaginable
n^n	10 bil.	85 dig.	201 dig.	744 dig.	unimaginable

μsecs since big bang: $\sim 10^{24}$

protons in the known universe: $\sim 10^{126}$

D. Harel, Algorithmics

COMPARING DIFFERENT ORDERS

How long would it take at 1 step per μsec?

	10	20	50	100	300
n^2	1/10,000 sec.	1/2500 sec.	1/400 sec	1/100 sec.	9/100 sec.
n^5	1/10 sec.	3.2 sec	5.2 min	2.8 hr	28.1 days
2^n	1/1,000 sec	1 sec	35.7 yr	400 trillion cent.	75 digit # of centuries
n^n	2.8 hr	3.3 trillion yr	70 digit # of centuries	185 digit # of centuries	728 digit # of centuries.

D. Harel, Algorithmics

A MILLION DOLLAR PROBLEM

A problem is of type P if it has a polynomial solution.

A problem is of type NP if, handed a solution to an instance of the problem, there is a polynomial time algorithm to check if it really is a solution.

e.g. factoring

QUESTION: P = NP?

> If P = NP, then the world would be a profoundly different place than we usually assume it to be. There would be no special value in "creative leaps," no fundamental gap between solving a problem and recognizing the solution once it's found. Everyone who could appreciate a symphony would be Mozart; everyone who could follow a step-by-step argument would be Gauss...
> — Scott Aaronson, MIT

NP-COMPLETE PROBLEMS

There is a (huge!) list of NP problems that "contain" all other NP problems, including:

Sudoku	Minesweeper
Battleship	Free Cell
Tetris	etc.

To prove P=NP, show any one of these problems has a polynomial solution.

To prove P≠NP, show any one of these problems has no polynomial solution.