O-Notation

O- (Big-Oh) notation provides a way of classifying functions according to their growth rates. For a given function $g(n)$, we denote by $O(g(n))$ the set of functions:

$O(g(n)) = \{f(n)$: there exist positive constants c and n_0 such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0\}$.

We write $f(n) = O(g(n))$ to indicate that $f(n)$ is a member of $0(g(n))$.

$\star$ Is the statement $\frac{1}{2}n^2 - 3n = O(n^2)$ true?

Is the statement $n = O(n^2)$ true?

Θ-Notation

The $Theta$-notation asymptotically bounds a function from above and from below. The definition is:

$\Theta(g(n)) = \{f(n)$: there exist positive constants c_1, c_2 and n_0 such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0\}$.

$\star$

⋆ Is the statement $\frac{1}{2}n^2 - 3n = \Theta(n^2)$ true?

Is the statement $n = \Theta(n^2)$ true?

Ω-Notation

The *Omega*-notation asymptotically bounds a function from below. The definition is:

$\Omega(g(n)) = \{f(n)$: there exist positive constants c and n_0 such that $0 \leq c_1 g(n) \leq f(n)$ for all $n \geq n_0\}$.

o-Notation

o- (Little-Oh) notation is like "$<$" while O notation is like "$\leq$". The actual definition is:

$o(g(n)) = \{f(n)$: For any positive constant c, there exist n_0 such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0\}$.

⋆ Is the statement $2n = o(n^2)$ true?

Is the statement $2n^2 = o(n^2)$ true?

The following table summarizes the definitions that will concern us.

Definition 1 *Let f and g be two non-negative functions that map the non-negative integers to the non-negative integers.*

We say that $f(n)$ is:	Mean that $f(n)$ grows:	Write:	If:
little-oh of $g(n)$	more slowly than $g(n)$	$f(n) = o(g(n))$	$\lim_{n\to\infty} f(n)/g(n) = 0$
big-oh of $g(n)$	no faster than $g(n)$	$f(n) = O(g(n))$	there exist some c, $n_0 > 0$: for all $n > n_0, f(n) \leq cg(n)$
theta of $g(n)$	about as fast as $g(n)$	$f(n) = \Theta(g(n))$	$f(n) = O(g(n))$ and $g(n) = O(f(n))$
approximately equal to $g(n)$	as fast as $g(n)$	$f(n) \approx g(n)$	$\lim_{n\to\infty} f(n)/g(n) = 1$
omega of $g(n)$	no slower than $g(n)$	$f(n) = \Omega(g(n))$	$g(n) = O(f(n))$

The definitions in the table give precise meanings to informal statements such as: $f(n)$ grows no faster than $g(n)$. A great feature of big-Oh notation is that it allows one to sweep under the rug inessential terms of a function and focus on the "dominant term.

The definition of big-Oh notation (second row of the table)

is designed to compare the growth rate of functions when constants are ignored. In contrast, the fourth row of the table defines notation that is useful when one wants to compare functions while taking constants into account. That is, if $f(n)$ is approximately equal to $g(n)$, the dominant terms have to be the exactly the same (including constants).

Limits are used in some of the definitions in the table. Recall that $\lim_{n\to\infty} f(n) = l$ if and only if for all $s > 0$ there exists some $n_0 > 0$ such that for $n > n_0$, $(l + s) > f(n) > (l - s)$.

$\star$ $\sum_{i=1}^{n} i^2$ vs. n^3.

Since

$$\sum_{i=1}^{n} i^2 \leq n^3,$$

it follows that

$$\sum_{i=1}^{n} i^2 = O(\quad)$$

We can look at this another way, which will be useful in

other examples later. We first show that:

Claim 1 *If* $\lim_{n\to\infty} f(n)/g(n) = r$ *then* $f(n) = O(g(n))$.

⋆ Proof:

We now apply this claim to show that $\sum_{i=1}^{n} i^2 = O(n^3)$:

⋆

Question: Why not just define $f(n) = O(g(n))$ if and only if $\lim_{n\to\infty} f(n)/g(n) = r$ for some $r \geq 0$?

Answer: If the above relation is true for a given pair of expressions, then $f(n) = O(g(n))$ is indeed always true. But there are some expressions where $f(n) = O(g(n))$ is true by the formal definition of O notation, but for whom the above relation is not true.

For example, consider the functions

$$f(n) = \begin{cases} 2n, & \text{if } n \text{ is even} \\ n, & \text{if } n \text{ is odd} \end{cases}$$

$$g(n) = \begin{cases} n, & \text{if } n \text{ is even} \\ 2n, & \text{if } n \text{ is odd} \end{cases}$$

⋆ Is $f(n) = \Theta(g(n))$?

Does $\lim_{n\to\infty} f(n)/g(n)$ exist?

Claim 2 *If* $\lim_{n\to\infty} f(n)/g(n) = r > 0$ *then* $f(n) = \Theta(g(n))$.

⋆ Proof:

From this claim, we see that in fact $\sum_{i=1}^{n} i^2 = \Theta(n^3)$.

$\star$ n vs. n^2.

$\star$ $n^2 + 2n$ vs. $2n^2 + n$

$\star$ n^2 vs. $n \log n$

These functions arise frequently in analyzing algorithms. Consider for example two sorting algorithms, Insertion Sort vs. Quick Sort. The expected number of comparisons for each sorting technique, assuming that the input is randomly permuted, is as follows:

Insertion Sort : $c_i n^2 + \text{o}(n^2)$.

Quicksort : $c_q n \,\text{logn} + \text{o}(\text{nlogn})$.

We want to compare these growth rates and focus on the dominant term of each function. (For this purpose, the constants c_i and c_q are unimportant.)

Let $f(n) = c_q n \lg n$ (where $\lg n = \log_2 n$).

Let $g(n) = c_i n^2$.

To compare growth rates, we can consider the limit of their ratios.

Intuitively we know that $\lg n$ grows much more slowly than n and so we expect that $\lim_{n\to\infty}[\lg n/n] = 0$. To prove this, we can apply l'Hopital's Rule.

L'Hopital's rule:

If the functions f and g are differentiable, $\lim_{n\to\infty} f(n) = \lim_{n\to\infty} g(n) = \infty$*, and* $\lim_{n\to\infty} f'(n)/g'(n)$ *exists, then* $\lim_{n\to\infty}$ *f(n)/g(n)* $= \lim_{n\to\infty}$ *f'(n)/g'(n).*

Now, getting back to Quicksort vs. Insertion sort, we have

⋆

Claim 3 *If g(n) goes to infinity and f(n) = o(g(n)), then* $2^{f(n)} = o(2^{g(n)})$.

⋆ Proof:

⋆ n vs. 2^n

In this lecture, facts about limits were used, both to define big-Oh and related notation, and to reason mathematically about the relative growth rates of functions. We will use O-notation extensively in this course, but almost always to compare very simple functions. So we will rarely need to go back to the "first principles" covered in this lecture. It is sufficient in this course to have an reasonable intuitive feel for limits – for example, you "know" that $\lim_{n\to\infty} \log n/n$ is zero (even you're not comfortable with l'Hopital's rule).

Another thing to remember is that if you're not sure how

to compare the growth rates of two functions using limits, you can always just evaluate the functions for a few values of n (e.g. $n = 10, 100, 1000$). From these values, you can often tell accurately if one function grows faster than the other. Try this when doing the following exercise, if you get stuck.

Exercise: Order the following functions according to their growth rates from slowest to fastest:

a) 2^{2n}	f) n $\lg n$
b)$\lg n$	g) $2^{n!}$
c) $\sqrt{n}$	h)2^n
d) n^n	i) $\lg(\lg n)$
e)$\lg^2 n$	j) $n!$
	k) 2^n

The following fact may be useful if you apply the limit methods described in this lecture to solve this problem. (See also the comments below in terms of other approaches.) Notice how big-oh notation is used in Stirling's formula.

Stirling's Formula: $n! = \sqrt{2\pi n}(n/e)^n(1 + O(1/n))$.

⋆

a) 2^{2n}

b)$\lg n$

c) $\sqrt{n}$

d) n^n

e)$\lg^2 n$

f) n $\lg n$

g) $2^{n!}$

h)2^n

i) $\lg(\lg n)$

j) $n!$

k) 2^n