Slide 1

Administrivia

- Reminder: Homework 7 due Monday.

Slide 2

Order of Magnitude of Functions, Review/Recap

- Often useful in analysis of algorithms to get a sense of how algorithm behaves (with regard to running time, memory needed, etc.) as size of input increases. Usually enough to get rough estimate — “order of magnitude” for functions, analogous to order of magnitude for numbers.
- Intuitive idea — classify according to “shape”.

Slide 3

Order of Magnitude of Functions, Continued

- Formal definition: $f = \Theta(g)$ (f and g have the same order magnitude) iff there are positive constants n_0, c_1, c_2 such that for all $x \geq n_0$

  $$c_1 g(x) \leq f(x) \leq c_2 g(x)$$

  Example from last time: $f(x) = x - 10$, $g(x) = 3x + 2$. We came up with $c_1 = 1/1000$, $c_2 = 1000$, $n_0 = 100$. (Plot with `gnuplot`.)

- Of course this is tedious to apply, so people have come up with some general rules. But first . . .

Slide 4

"Big-O Notation"

- The $O(f(N))$ you see in computer science is similar, but it's a "less than or equal" rather than a "strictly equal" — i.e., $f(N) = O(g(N))$ means f's order of magnitude is no bigger than g's (and might be less).

  Formally, true iff there are positive constants n_0 and c such that for all $x \geq n_0$

  $$f(x) \leq cg(x)$$

- Interesting (?) to observe that Θ is an equivalence relation, and O is a partial ordering.

Slide 5

Order of Magnitude of Functions, Continued

- So we have a way to compare orders of magnitude of functions, with an "equals" (Θ) and a "less-than-or-equal-to" (O).
- In general, function's order of magnitude determined by fastest-growing term. Some categories of interest:
  - x^2 grows faster than x, x^3 faster than x^2, etc. x^2 and cx^2 "the same".
  - $\log_b x$ grows more slowly than x.
  - b^x grows faster than all polynomials.
  - x^x grows faster than all b^x.

Slide 6

Graphs — Overview

- In some contexts, "graph" means a plot of a function, other pictorial representation of data.
- In other contexts, it's an abstract idea meant to represent relationships among a set of things. Examples:
  - Hasse diagrams of chapter 4.
  - Airline route maps.
  - Simplified maps showing driving distances between cities.

  Common idea — set of things (set elements, cities) and a notion that some pairs of them are connected somehow. Details we don't care about have been "abstracted out".

Slide 7

Graphs — Definition

- Formal definition (undirected graph):
  - Nonempty set N of nodes (vertices).
  - Set A of arcs (edges).
  - Function $f : A \rightarrow \{\{x, y\} | x, y \in N\}$ (unordered pairs of nodes).

  Notice that we can have "loops" and also "parallel arcs".
- Variations/extensions:
  - Directed graph — edges are ordered pairs (i.e., "one-way").
  - Labeled graph — each vertex has some associated info ("label").
  - Weighted graph — each edge has some associated info ("weight").

Slide 8

Graphs — Terminology

- Adjacent nodes (arc from one to the other).
- Loop, parallel arc.
- Simple graph (no loops or parallel arcs).
- Complete graph (every pair of nodes adjacent).
- Path (sequence of arcs), connected graph.
- Cycle, acyclic graph.

Slide 9

Minute Essay

- Which of the following functions are $O(N^2)$?

  $g(N) = 100N^2 + N - 1000$

  $h(N) = N^3$

- Which of the following functions are $O(2^N)$?

  $f(N) = 2^N - 5$

  $h(N) = N!$

Slide 10

Minute Essay Answer

- $O(N^2)$?

  $g(N) = 100N^2 + N - 1000$ — yes

  $h(N) = N^3$ — no

- $O(2^N)$?

  $f(N) = 2^N - 5$ — yes

  $h(N) = N!$ — no