Slide 1

Administrivia

- Reminder: Quiz 6 Tuesday.
- Reminder: Homework 7 due today.
- Next homework should be on the Web soon. I will send mail. Due last day of class.

Slide 2

Order of Magnitude of Functions

- By now you've probably heard "this is an $O(N)$ algorithm", etc., many times. Here we'll define it formally.
- First: When we talked about analysis of algorithms (chapter 2), we came up with estimates of "total work" of the algorithm as a function of size of input ("problem size"). Useful and interesting, but a bit fine-grained — what we usually care about is behavior as problem size gets very big.
- So — idea is to come up with an "order of magnitude" for functions, analogous to "order of magnitude" for numbers. If the functions for two algorithms have the same order of magnitude, the functions are in some sense about equally fast/slow.
- Example: If you have two algorithms for processing an image with N pixels, one that takes time proportional to $1000N$ and one that takes time proportional to time N^2, which do you pick? (Does the size of N matter?)

Slide 3

Order of Magnitude of Functions, Continued

- How to determine an order of magnitude for functions?

  If we look at graphs of functions, we might notice that we can classify them into groups based on their "shape".

  For nondecreasing functions, we also notice that some shapes "grow" faster than others.

  (Compare x^2, $10x^2$, x^3, etc.)

- Idea is that we want functions that have the same shape to have the same order of magnitude.

Slide 4

Order of Magnitude of Functions, Continued

- Formal definition:

  Write $f = \Theta(g)$ to mean that f and g have the same order magnitude.

  Define to be true iff there are positive constants n_0, c_1, c_2 such that for all $x \geq n_0$

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

  In other words, these functions are roughly proportional to each other.

- Can guess values c_1, c_2 and more or less show that they work by plotting resulting curves — but to really show that the definition holds, must do algebra to show. (Example.)

Slide 5

"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 6

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 7

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 8

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 9

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 10

Isomorphic Graphs

- What we care about is the relationship between nodes and arcs, not exact visual representation.
- Can formalize this as “isomorphism” — two graphs are isomorphic if one is just a “relabeling” of the other.
- Formal definition is in terms of one-to-one functions, one from nodes of graph G_1 to nodes of graph G_2 and one from arcs of graph G_1 to arcs of graph G_2. Idea is that if an arc connects two nodes in G_1, the corresponding arc in G_2 connects the corresponding nodes.

Computer-Friendly Representation of Graphs

- For humans, representing graphs pictorially usually works well. For computers, other representations work better.
- Key idea is to come up with a way to represent the essential information — set of nodes and which ones are connected.

Slide 11

Adjacency Matrices

- Idea is to put the n nodes in some (arbitrary) order and define an n-by-n matrix A such that A_{ij} is the number of arcs connecting node i and node j.
- For an undirected graph, what property does this matrix have? that it might or might not have for a directed graph?
- Variation: For a weighted graph with no parallel arcs, we could let A_{ij} be the weight of the arc from node i to node j.

Slide 12

Adjacency Lists

Slide 13

- Idea is to again put n nodes in some arbitrary order, but rather than a matrix define an array of n lists, one for each node, with the list for node i containing all nodes j that are adjacent to node i. Parallel arcs mean "duplicate" entries.

Adjacency Matrix Versus Adjacency List

Slide 14

- Which uses less space?
- Which makes it faster to answer the question "is node i adjacent to node j?"

Slide 15

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 16

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