# Algorithms and Computation in Signal Processing

special topic course 18-799B
spring 2005
3rd Lecture Jan. 18, 2005

Instructor: Markus Pueschel

TA: Srinivas Chellappa

# Miscellaneous

- Assignments on Website, due next Tuesday before class
- PC for programming assignments
- Project

# Asymptotic Analysis (cont'd)

- Problems with fixed upper bound on input size
  - The complexity of multiplying 2 matrices of dimensions smaller than 1000 x 1000 has complexity O(1)
- Multiple parameters (blackboard)
  - Mat-mat multiplication
  - Polynomial multiplication

# Asymptotic Analysis: Remarks

- Asymptotic runtime analysis works since it is independent of the exact runtime of the elementary steps counted (including memory latencies)
- Complexity of a problem is usually stated using “O” since every algorithms provides an upper bound, but lower bounds are hard to get
- People often talk about “complexity of an algorithm” which, in a strict sense, is wrong
- Problem: asymptotic analysis gives only an asymptotic idea of the runtime, but in real implementations the constants (and the algorithmic structure) matter

# Cost Analysis

# Refined Analysis for Numerical Problems

- Goal: determine exact static “cost” of algorithms
- Approach (use mat-mat mult. as running example):
  - Fix an appropriate cost measure C, “what do I count”
  - Determine cost of algorithm as function C(n) of input size n, or, more general, of all relevant input parameters:

$$C(n_1,..,n_k)$$

  - Cost can be multi-dimensional

$$C(n_1,..,n_k) = (c_1,..,c_m)$$

- Exact cost gives a more precise idea of runtime but not the exact runtime

# Cost Analysis

- Examples
  - Count additions and multiplications for flop rate
  - Mat-mat mult.
  - Polynomial mult.
- Solving recurrences
  - Great book: Graham, Knuth, Patashnik, “Concrete Mathematics,” 2nd edition, Addison Wesley 1994
  - Blackboard

# For Publications

- A problem has a complexity
- An algorithm has a cost (e.g., operations count, runtime, memory requirement, area requirement in hardware)
- Cost=runtime can only be analyzed asymptotically
- In a precise sense, an algorithm does not have a complexity

| Problem | Complexity<br>Runtime compl. (asympt.) |
|---|---|
| Algorithm | Cost<br>Runtime (asymptotic) |

In research/writing/publications:

If your contribution is an algorithm, you have to analyze it. As follows:

1) state your cost/complexity measure (what you count);
2) compute the cost of the algorithm as precise as possible/necessary, at least asymptotically;
3) state what you know about the complexity of the problem you address (from theory, other algorithms, …)