ECS 122A
Algorithm Design and Analysis

Instructor: Qirun Zhang

Course slides (partially) adopted from the notes by David Luebke.

Agenda

- Asymptotic notations
- Merge sort

Course updates

- About homework
  - Homework 1 has 5 problems
  - Submit 5 separate solutions on gradescope (i.e., one for each problem)
- Prerequisite petition
  - Send me a reminder email next week

Why asymptotic notation?

- Asymptotic efficiency

Recap

- Simplifications
  - Ignore actual and abstract statement costs
  - *Order of growth* is the interesting measure:
    - Highest-order term is what counts
      - Remember, we are doing asymptotic analysis
      - As the input size grows larger it is the high order term that dominates

Upper Bound Notation

- We say InsertionSort's run time is $O(n^2)$
  - Properly we should say run time is *in* $O(n^2)$
  - Read O as "Big-O" (you'll also hear it as "order")
- In general a function
  - $f(n)$ is $O(g(n))$ if there exist positive constants c and n_0 such that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$
- Formally
  - $O(g(n)) = \{ f(n): \exists$ positive constants c and n_0 such that $f(n) \leq c \cdot g(n)\ \forall\ n \geq n_0$

Lower Bound Notation

- We say InsertionSort's run time is $\Omega(n)$
- In general a function
  - $f(n)$ is $\Omega(g(n))$ if $\exists$ positive constants c and n_0 such that $0 \leq c \cdot g(n) \leq f(n) \quad \forall\ n \geq n_0$

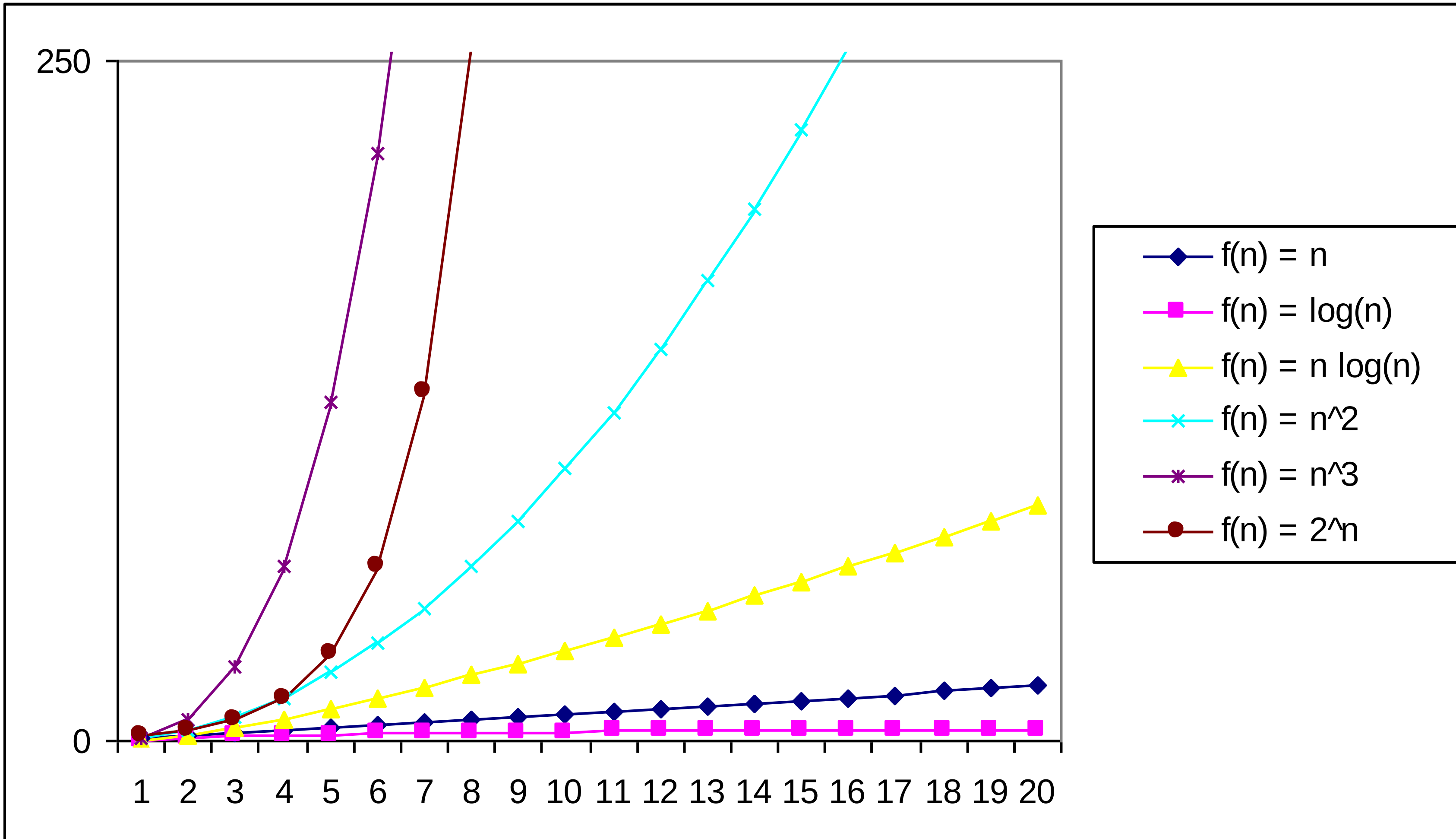
Asymptotic Tight Bound

- A function f(n) is $\Theta(g(n))$ if $\exists$ positive constants c_1, c_2, and n_0 such that

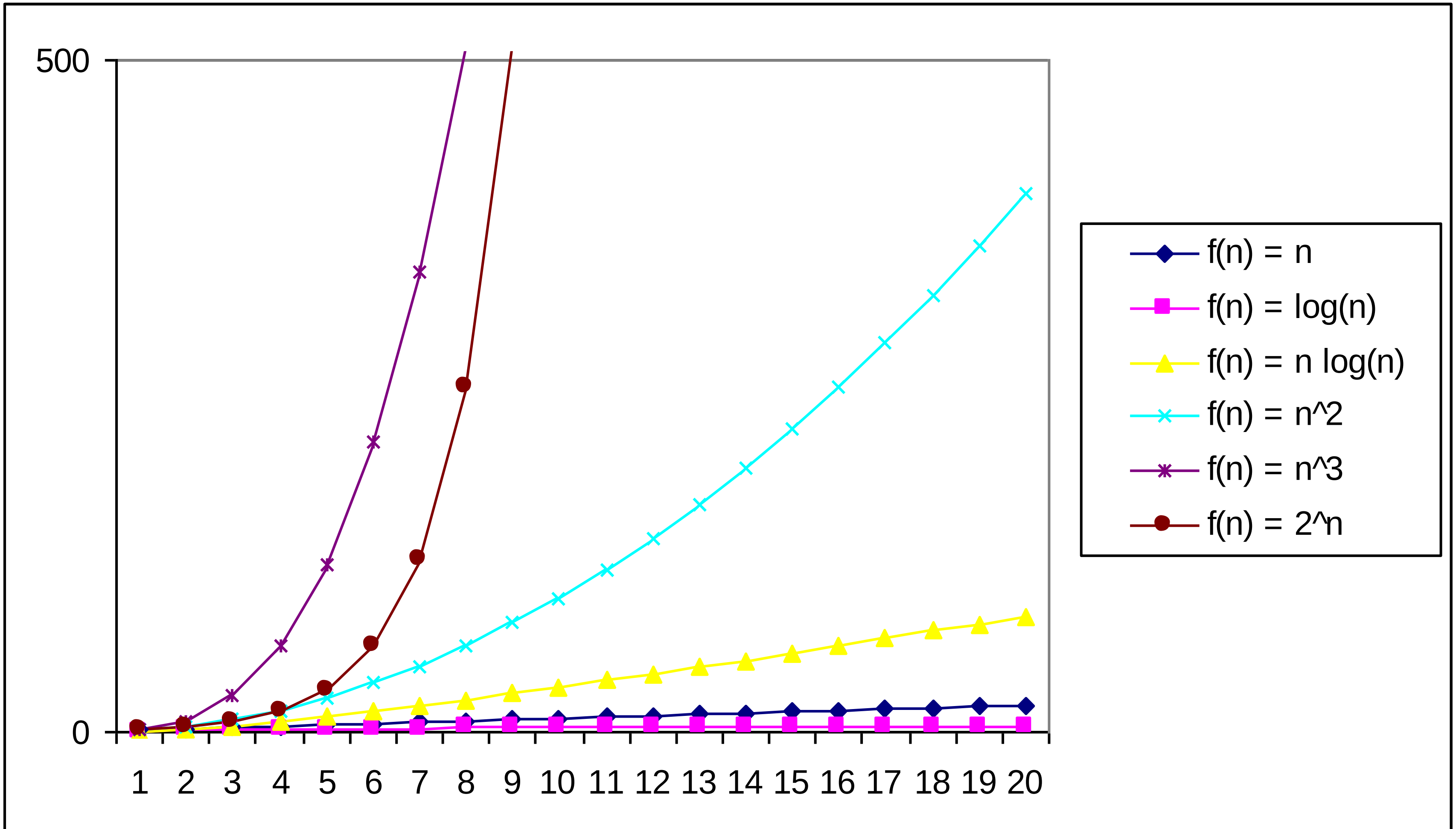
$$c_1\, g(n) \le f(n) \le c_2\, g(n) \;\forall\; n \ge n_0$$

- Theorem
  - f(n) is $\Theta(g(n))$ iff f(n) is both $O(g(n))$ and $\Omega(g(n))$

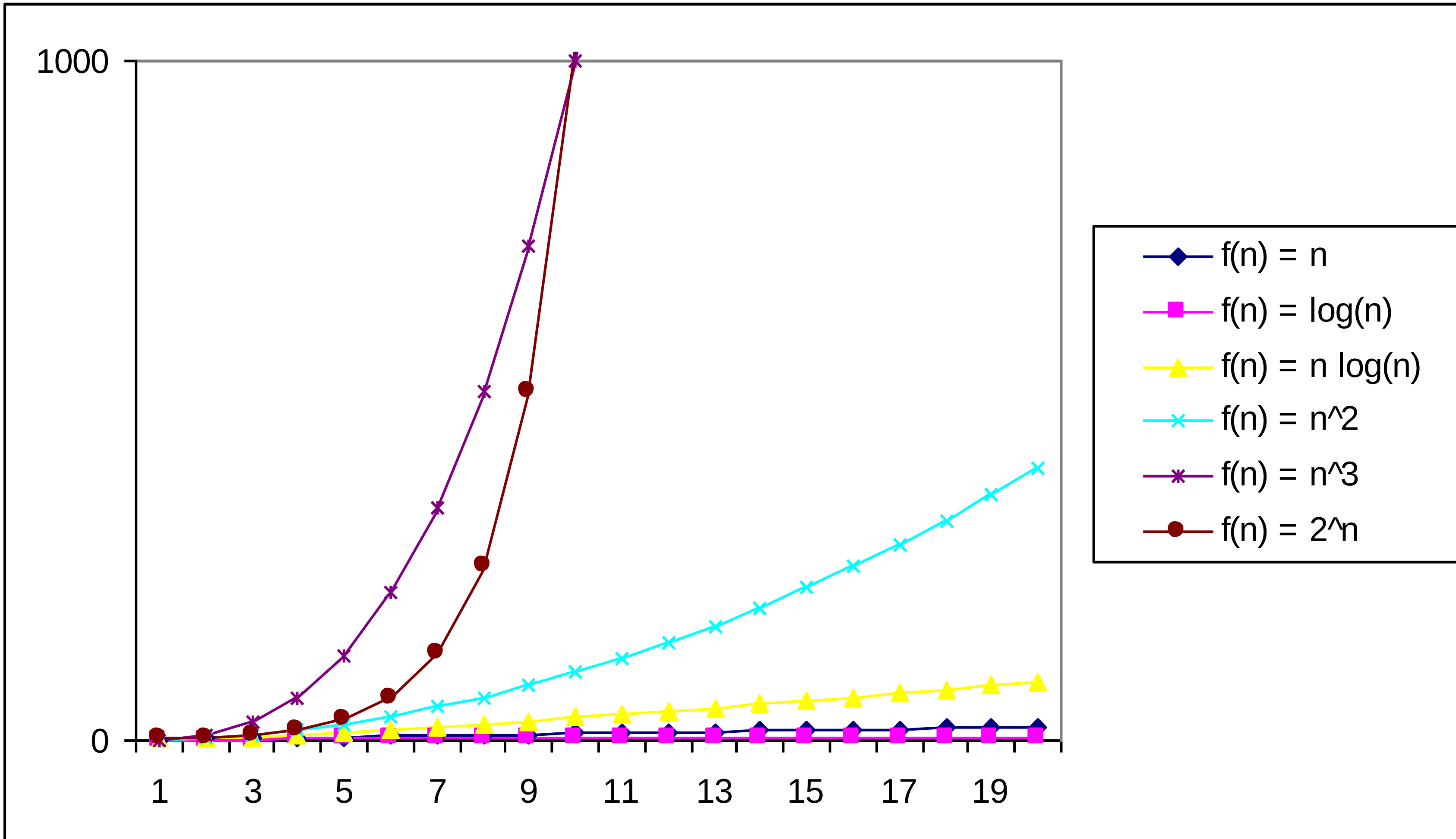
Practical Complexity



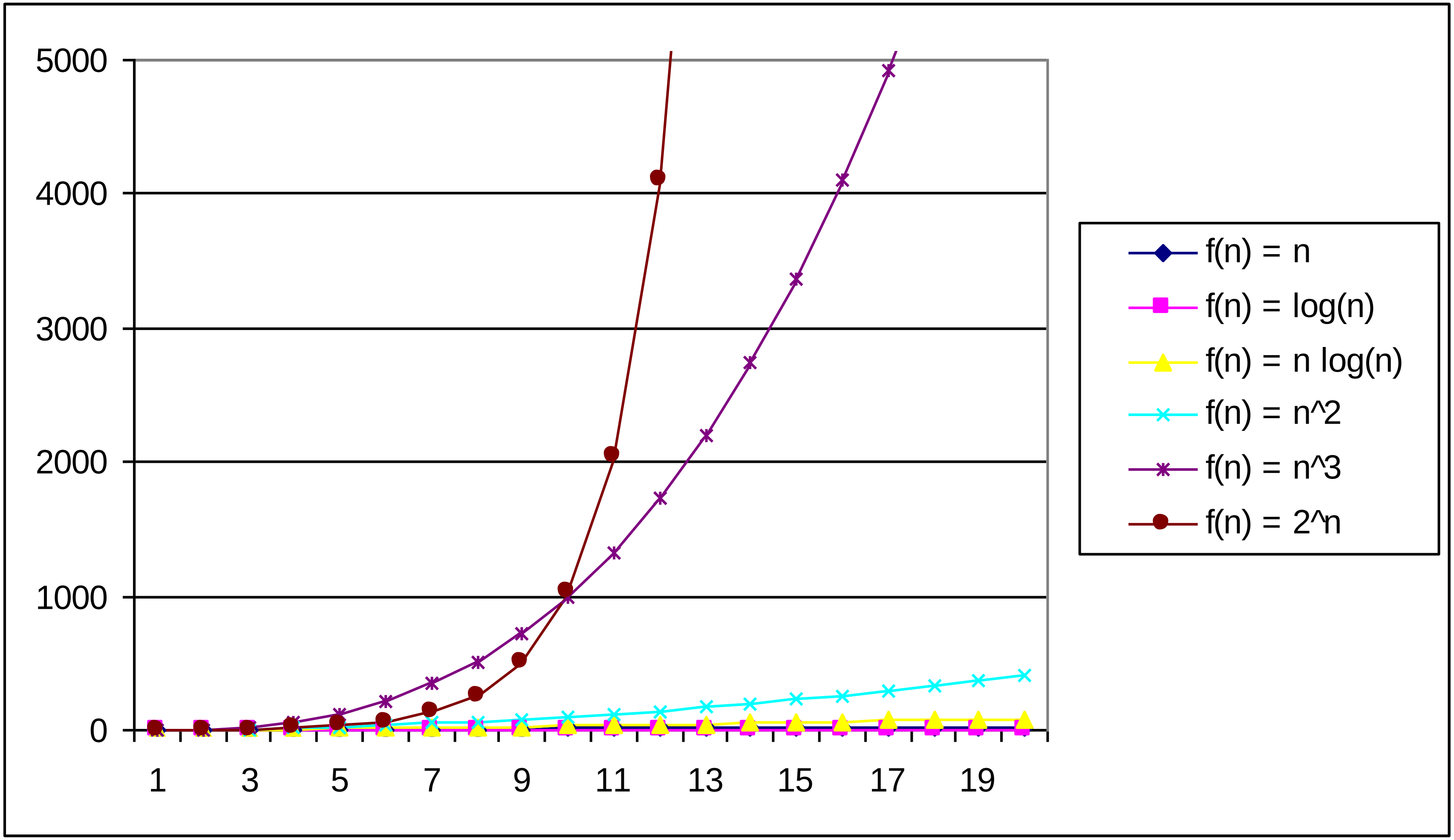
Practical Complexity



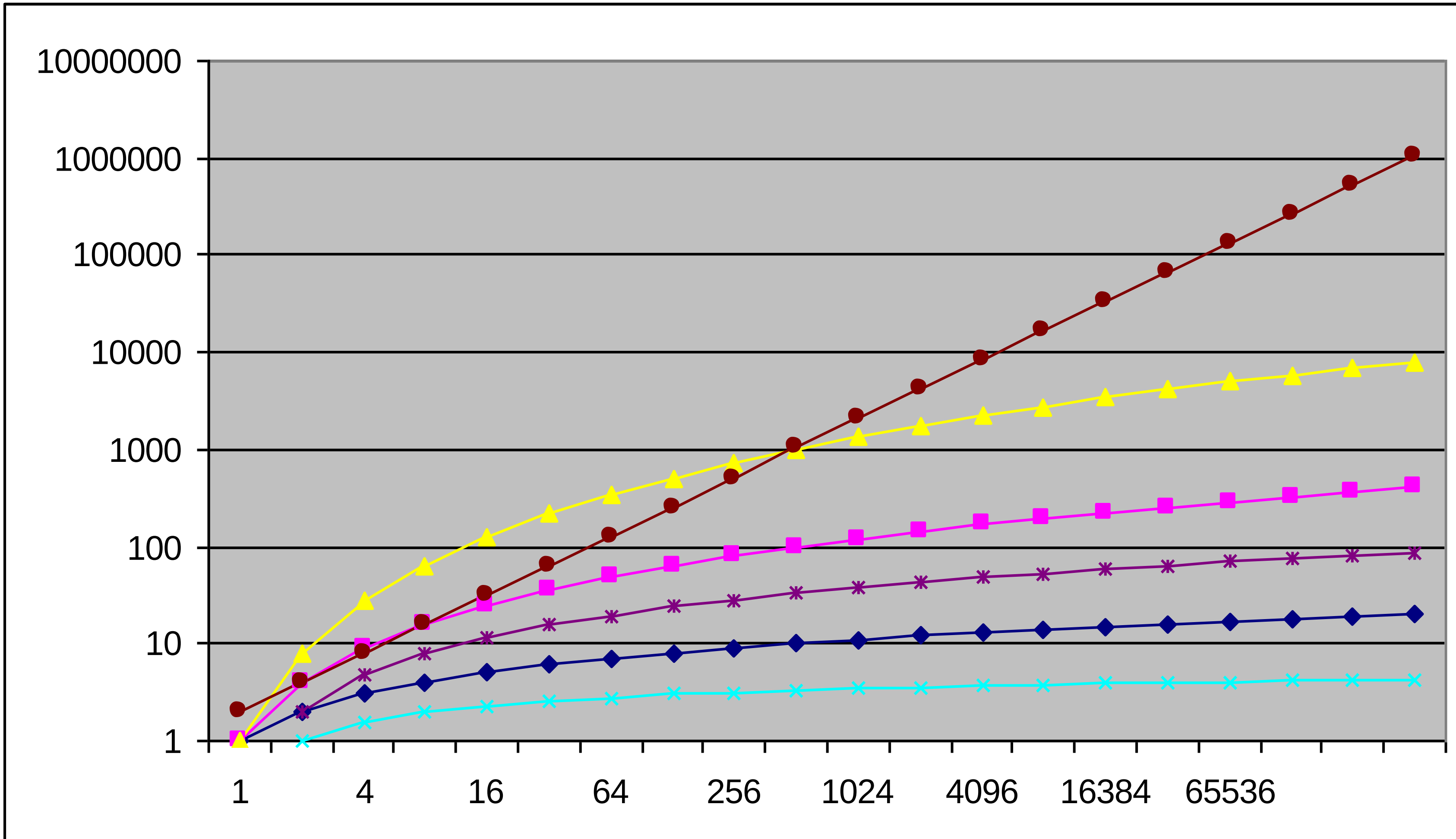
Practical Complexity



Practical Complexity



Practical Complexity



Other Asymptotic Notations

- A function f(n) is o(*g*(n)) if $\forall$ positive constants *c*, there exists n_0 such that
  $$f(n) < c\,g(n) \;\forall\; n \geq n_0$$
- A function f(n) is ω(*g*(n)) if $\forall$ positive constants *c*, *there exists* n_0 such that
  $$c\,g(n) < f(n) \;\forall\; n \geq n_0$$
- Intuitively,

  - o() is like <
  - ω() is like >
  - Θ() is like =
  - O() is like $\leq$
  - Ω() is like $\geq$

Summary

Growth	Terminology
$O(1)$	constant growth
$O(\log n)$	logarithmic growth
$O(\log^k n)$, for some $k \geq 1$	polylogarithmic growth
$o(n)$	sublinear growth
$O(n)$	linear growth
$O(n \log n)$	log-linear growth
$O(n \log^k n)$, for some $k \geq 1$	polylog-linear growth
$O(n^k)$ for some $k \geq 1$	polynomial growth
$\Omega(n^k)$, for every $k \geq 1$	superpolynomial growth
$\Omega(a^n)$ for some $a > 1$	exponential growth

Merge Sort

Merge Sort: Example

Analysis of Merge Sort

The End