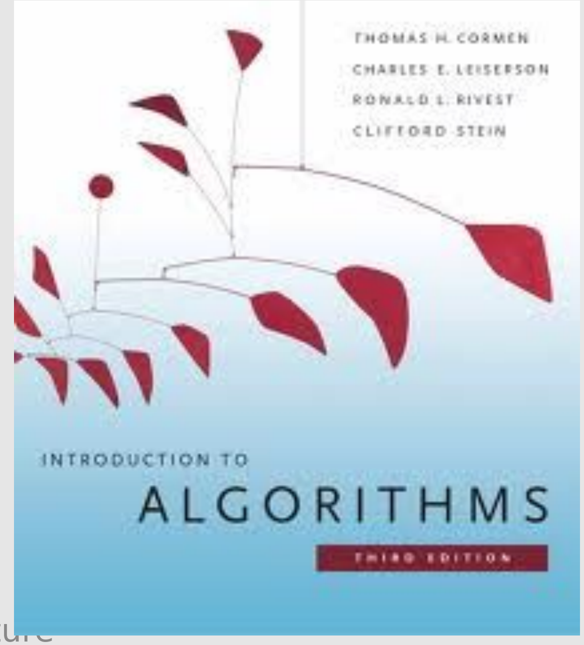
Algorithm and Data Structure Analysis (ADSA)

Lecture 1: Introduction

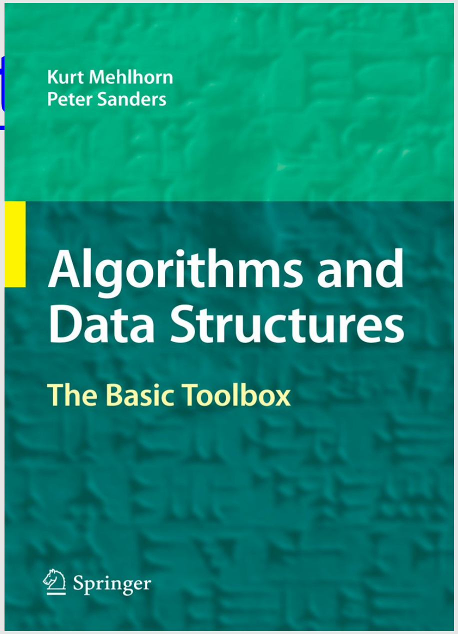
Course materials

- Textbook (optional): Introduction to Algorithms, T. Corman, Leiserson, Rivest, Stein.



Course materials

- Reference Book: K. Mehlhorn, P. Sanders: Algorithms and Data Structures, Springer, 2008
- http://www.mpi-inf.mpg.de/~mehlhorn/Toolbox.html



Let’s get started

Motivation

Why is this course important?

- Efficient data structures and algorithms are essential for successful computer applications
- We need efficient methods on how to store, manipulate data
- We need efficient algorithms to search them.
- Problems in computer science require efficient algorithms.

Topic of this course:

Basic data structures and algorithms with focus on their analysis

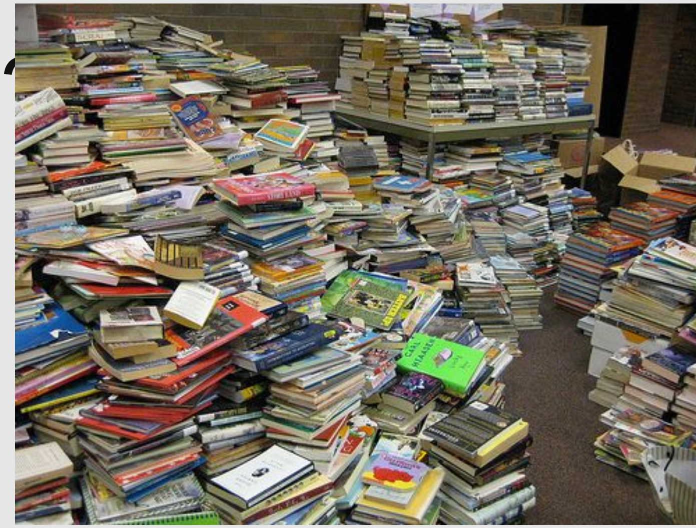
Goals

We want to have:

- Data structures that allow us to carry out operations as efficiently as possible
- Algorithms that solve problems as efficiently as possible.

Appetizer

Sorting: Why should I care?



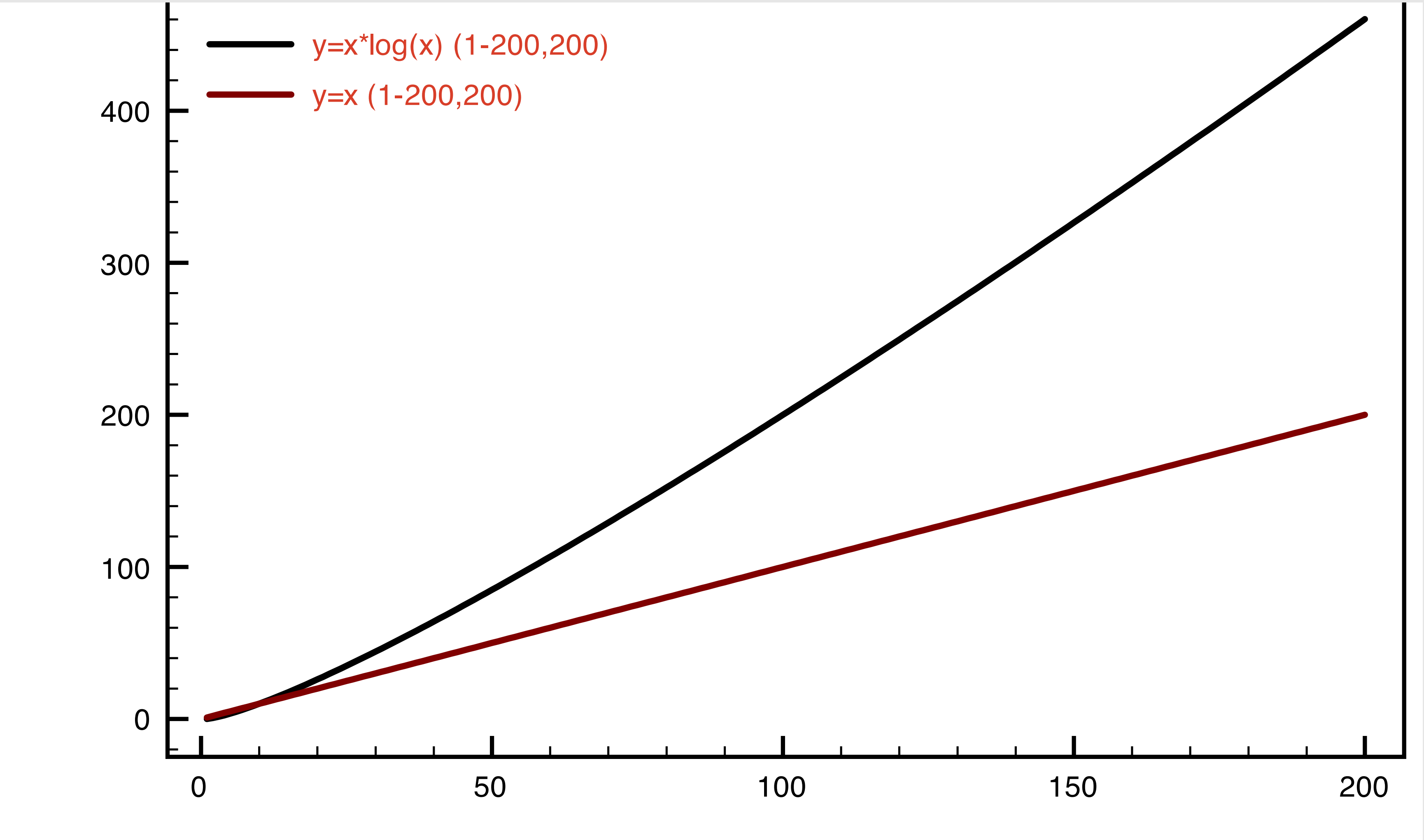
Which one do you prefer?

Efficiency

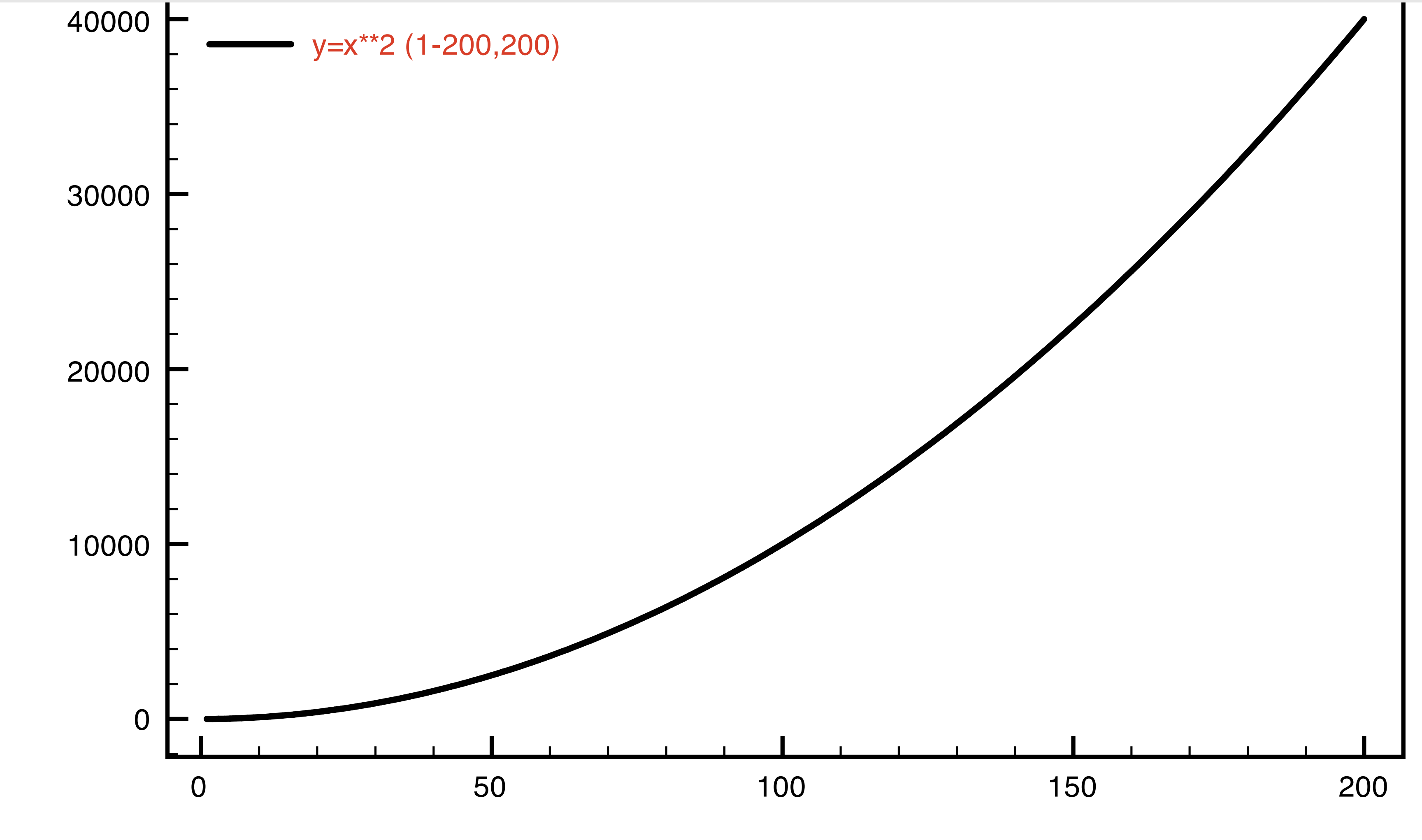
What’s our measure?

- Let n be the number of input elements (think of number of books)
- Measure time for operations on data structures and time to execute algorithms in dependence of n
- Consider orders of magnitude

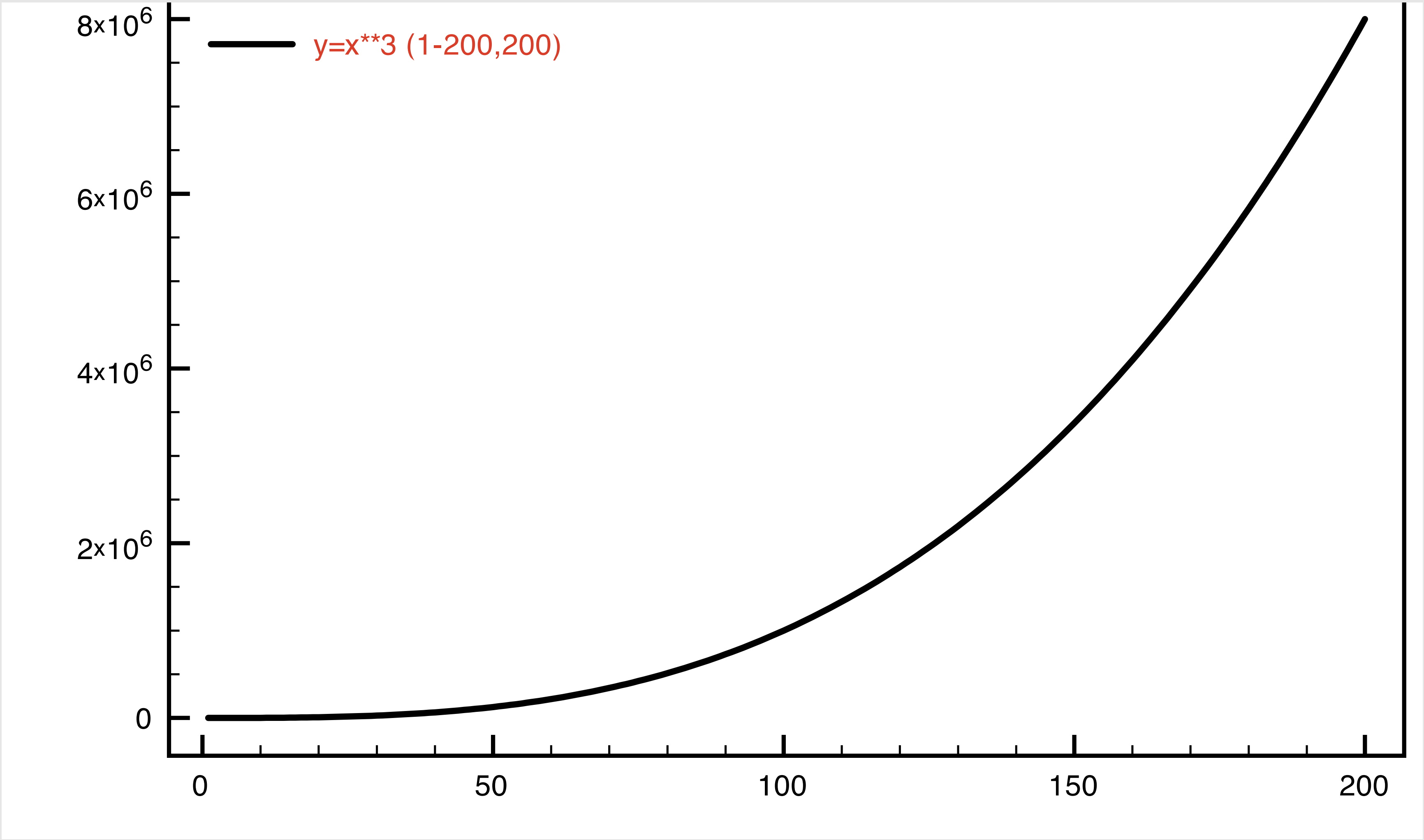
(Almost) Linear Runtime



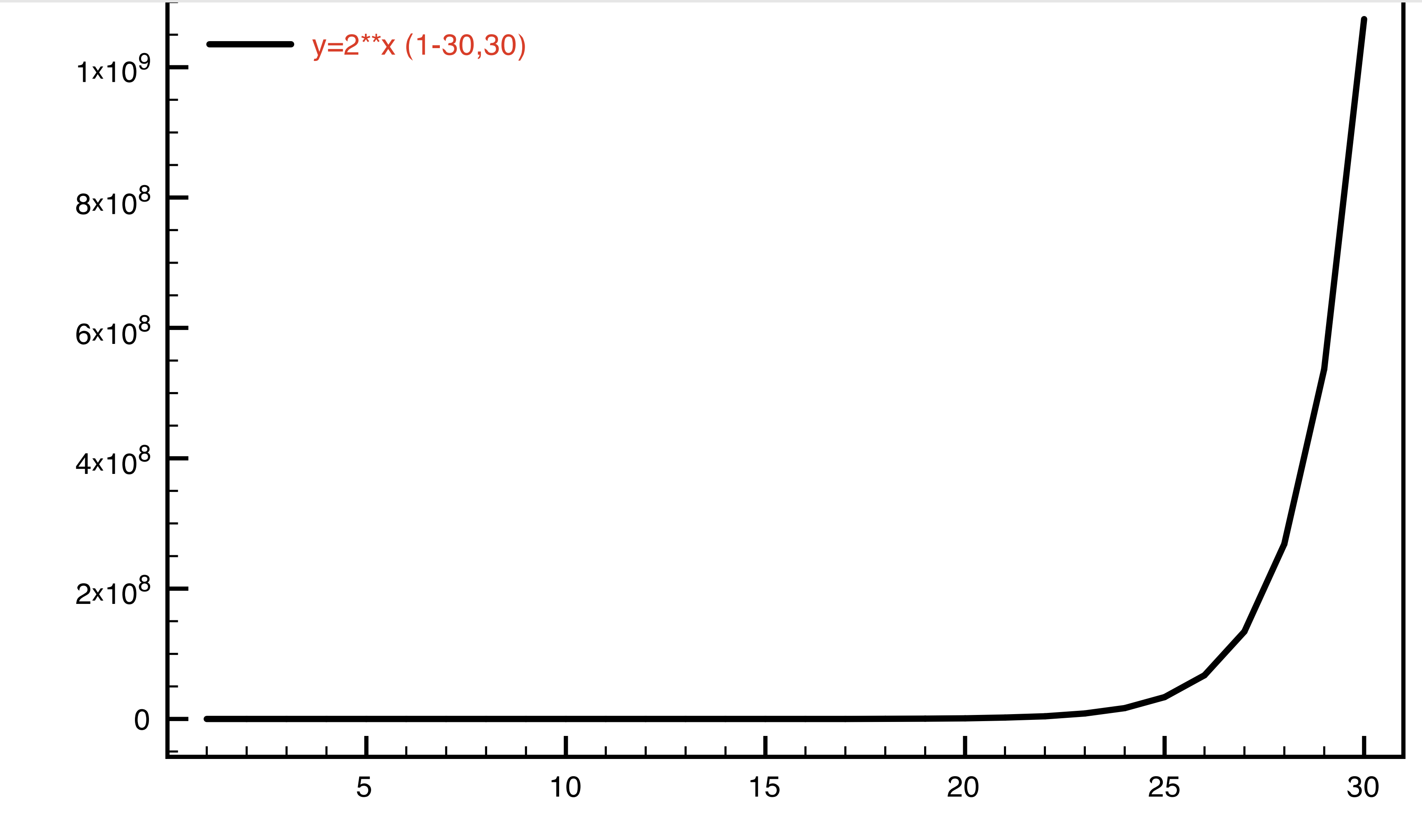
Quadratic Runtime



Cubic Runtime



Exponential Runtime



Complexity

n	n $\log_{10}$ n	n^2	n^3	2^n
10	10	100	1000	1.024
100	200	10.000	1.000.000	2^100
1000	3.000	1.000.000	1.000.000.000	2^1000
10000	40.000	100.000.000	10^{12}	2^10000

It's great to have algorithms that run in linear time or time n logn.

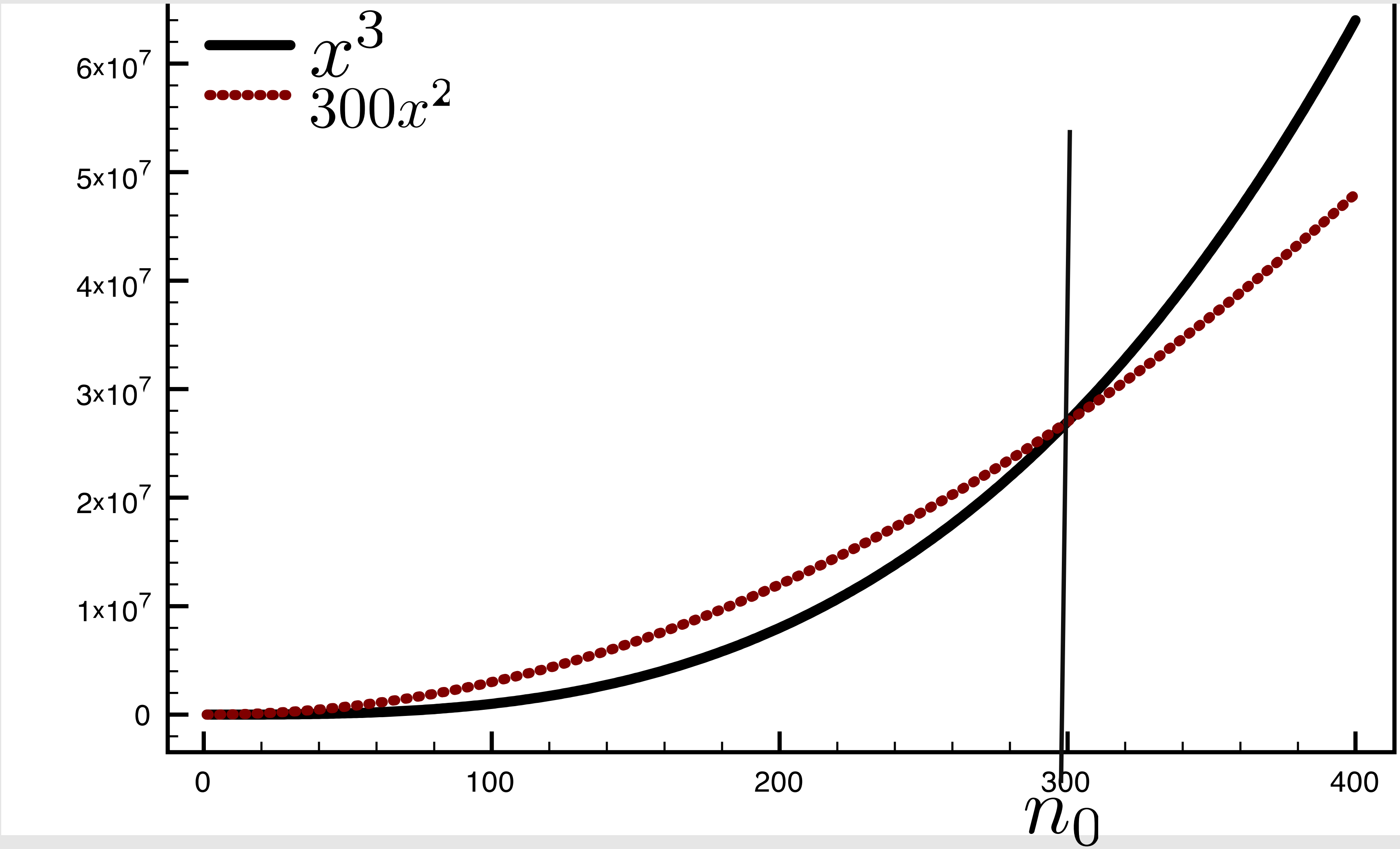
Many important problems have algorithms whose runtin is bounded by a small polynomial (e. g. n^2 or n^3)

For a wide class of important problems there is most probably no algorithm that runs in polynomial time.

Asymptotic Behavior

- We measure runtime as a function of the input size n.
- Define complexity depending on the asymptotic behavior.
- Want to have algorithms that solve a given problem and have low complexity.

Asymptotic Behavior



Landau Symbols

We want to measure computation times asymptotically

$$\mathrm{O}(f(n)) = \{g(n) : \exists c > 0 : \exists n_0 \in \mathbb{N}_+ : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\},$$
$$\Omega(f(n)) = \{g(n) : \exists c > 0 : \exists n_0 \in \mathbb{N}_+ : \forall n \geq n_0 : g(n) \geq c \cdot f(n)\},$$
$$\Theta(f(n)) = \mathrm{O}(f(n)) \cap \Omega(f(n)),$$
$$\mathrm{o}(f(n)) = \{g(n) : \forall c > 0 : \exists n_0 \in \mathbb{N}_+ : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\},$$
$$\omega(f(n)) = \{g(n) : \forall c > 0 : \exists n_0 \in \mathbb{N}_+ : \forall n \geq n_0 : g(n) \geq c \cdot f(n)\}.$$

Mehlhorn, Sanders (page 21)

We often write $h = O(f)$ instead of $h \in O(f)$ and $O(h) = O(f)$ instead of $O(h) \subseteq O(f)$.

Examples

$$5n : O(n), \Omega(n), \Theta(n), o(n \log n), \omega(\sqrt{n})$$

$$n^2 - n \log n : O(n^2), \Omega(n^2), \Theta(n^2), o(n^3), \omega(n \log n)$$

$$100n : O(n^2), \Omega(\sqrt{n}), \Theta(n), o(n \log n), \omega(\sqrt{n})$$

Right or Wrong

$5n \log n \in O(n \log n)$ Right

$5n \log n \in O(n^2)$ Right

$5n \log n \in \Omega(n^2)$ Wrong

$5n \log n \in o(n^2)$ Right

$5n \log n + n^2 \in O(n \log n)$ Wrong

$5n \log n + n^2 \in O(n^2)$ Right

Summary

- Efficient data structures and algorithms are crucial for successful computer applications.
- Measure runtime as a function of the given input size.
- Asymptotic behavior and complexity classes.
- Reading: Mehlhorn & Sanders ch 2.1