Implementing Selection Sort

Efficiency

Sorting: Putting Data in Order

- This sorting algorithm: selection sort
  - Pick smallest, put in place, repeat
  - **Simple**
  - **Slow on large data**
- Other approaches?
- Two categories:
  - Simple + slow: runtime **quadratic** in data size
  - Clever + fast: runtime **close to linear** in data size

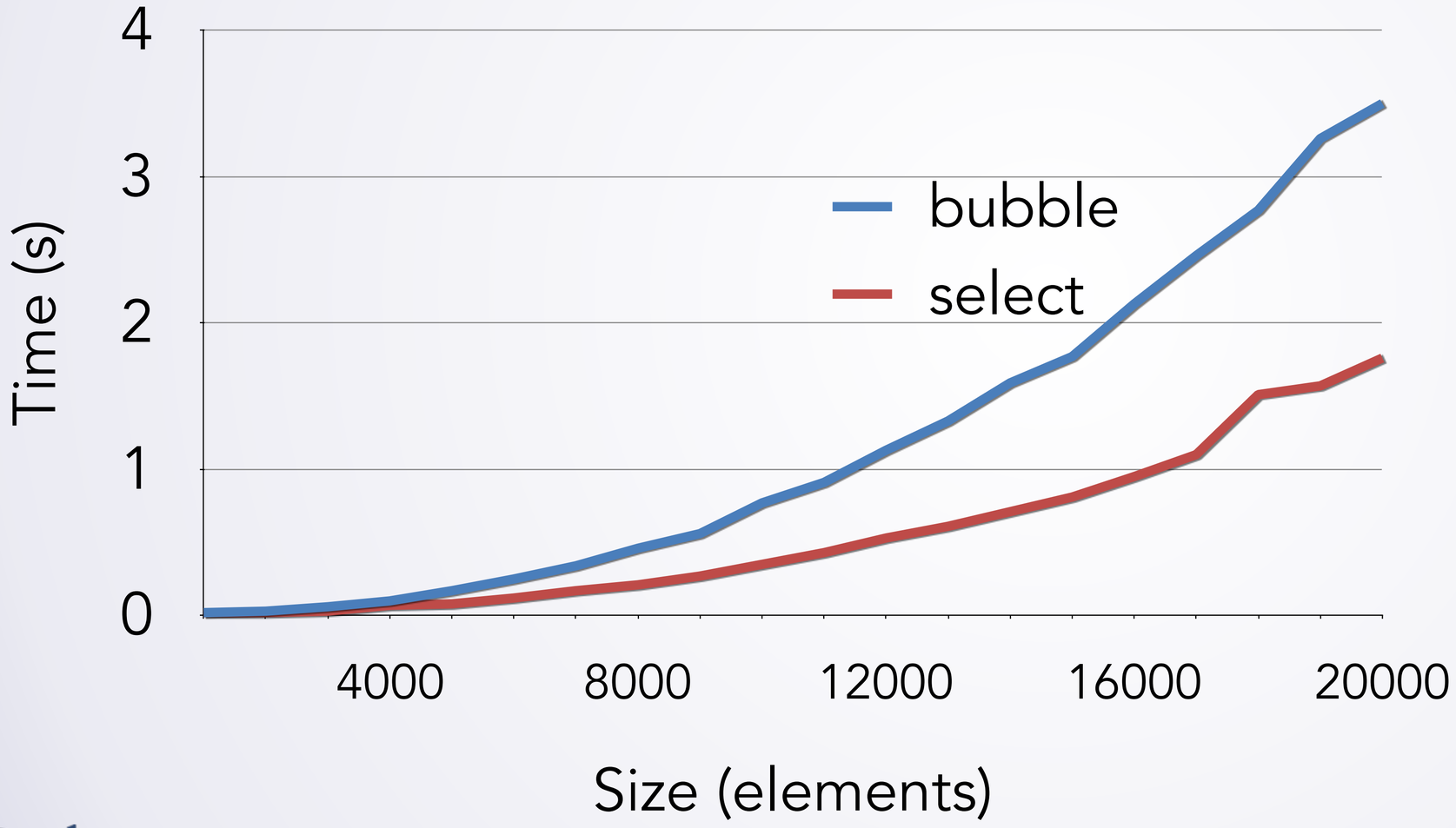
SortTimings Data

Using SortTimings.java for n^2 sorts

- Bubble and Selection have same general shape, algorithms are easy to understand

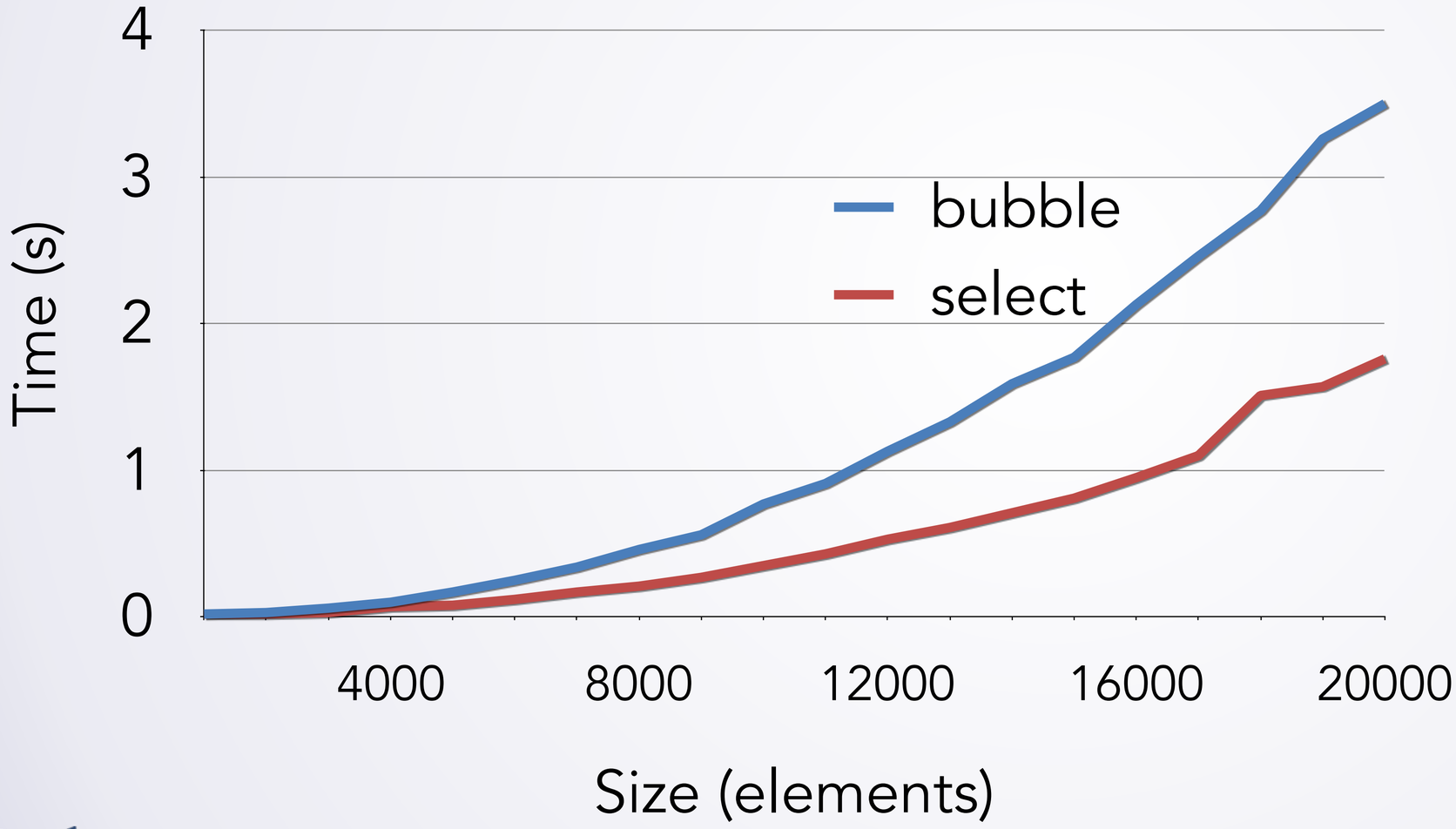
SortTimings Data

Less than 4 and 2 seconds for 20,000 strings



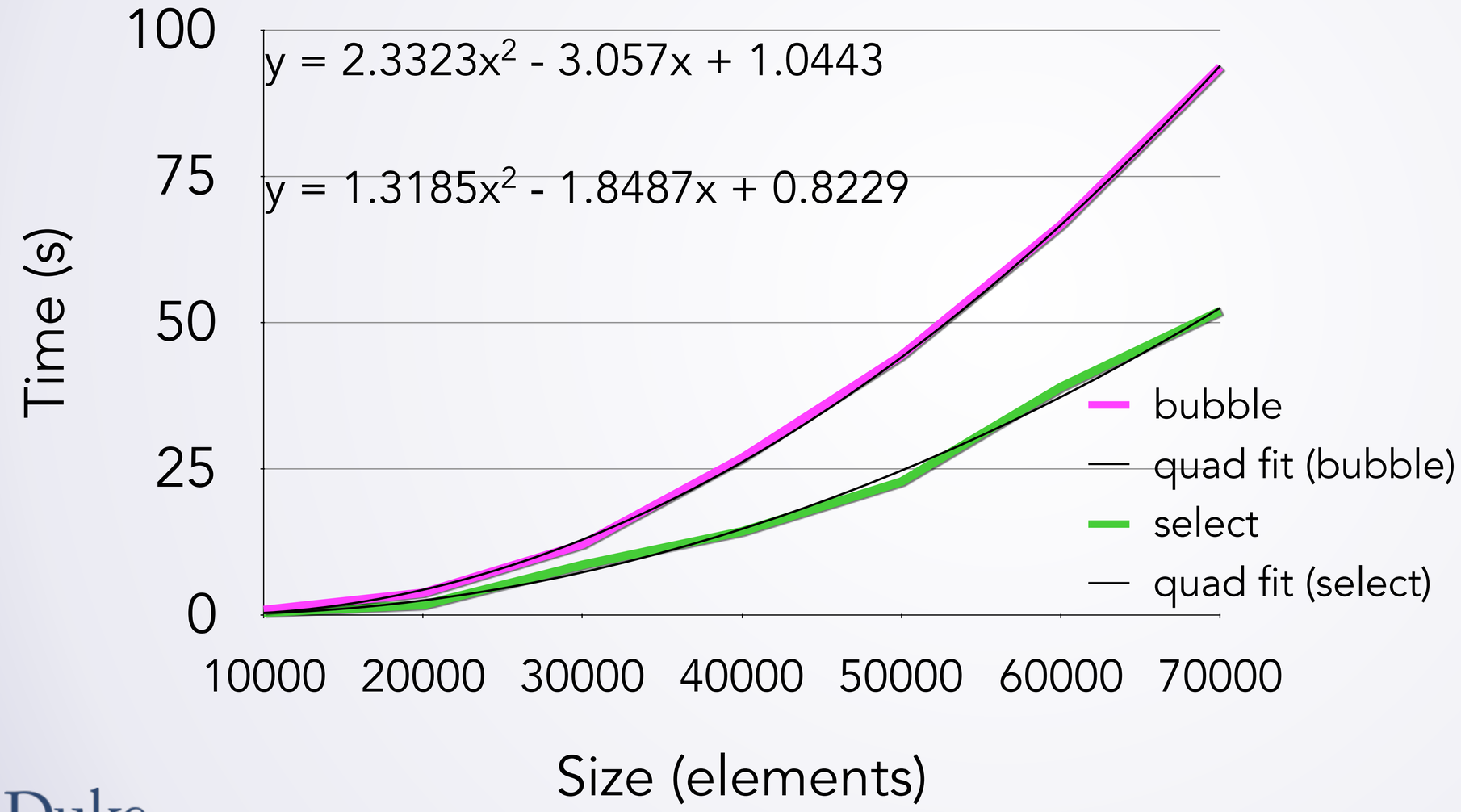
SortTimings Data

Perhaps acceptable for sorting small lists



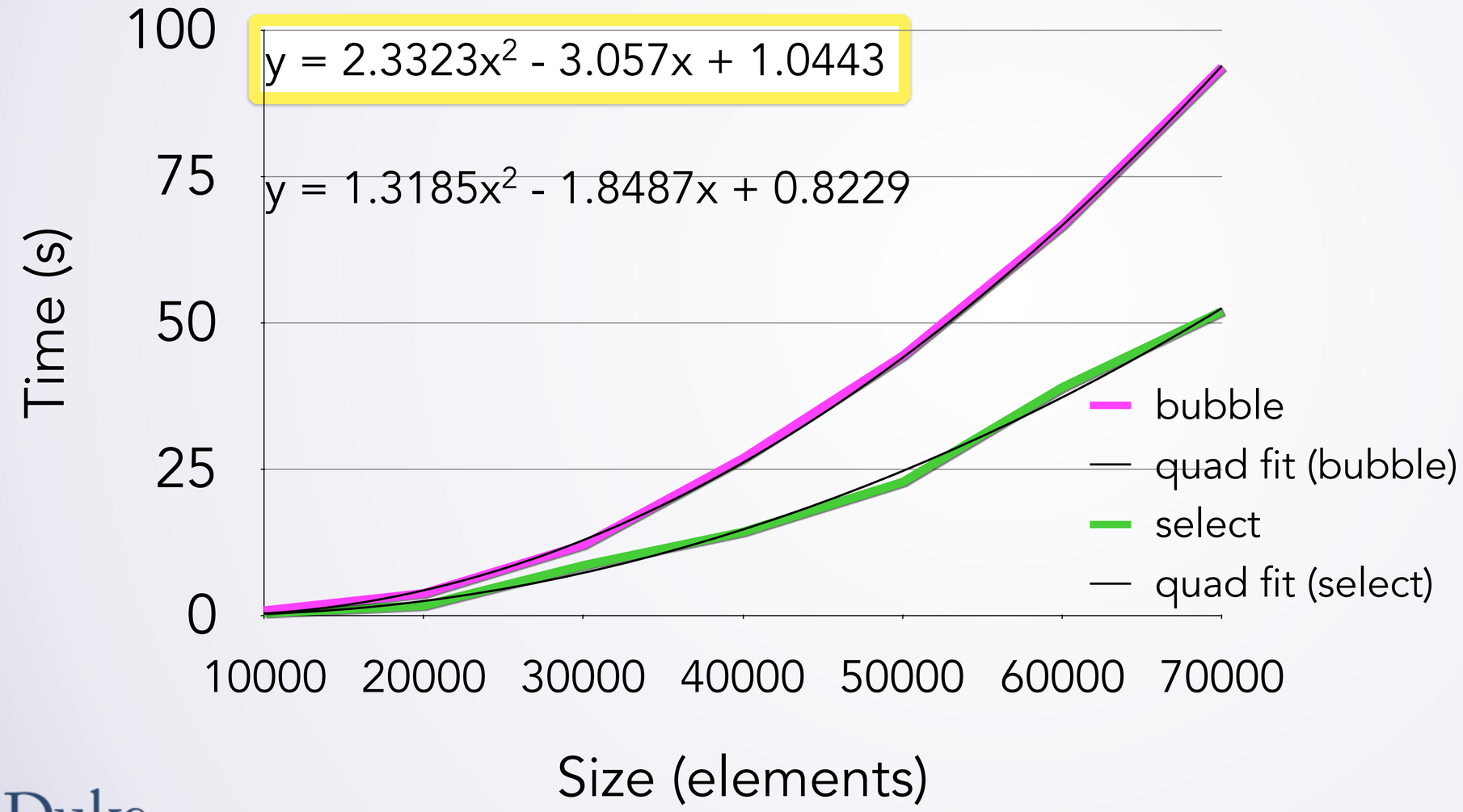
SortTimings Data

Sorting 10,000 to 70,000 Strings



SortTimings Data

Sorting 10,000 to 70,000 Strings



SortTimings Data

	Bubble sort	Collections.sort
1 million elements	6.4 hours	1 sec
1 billion elements	738 years	19 min

Sorting: Common

- Most languages: Sorting built in
  - Typically a very efficient sort
  - Needs to work with variety of types
    - Did author think about earthquakes?
    - How to do this? **Interfaces**
      - Comparable
      - Comparator
  - Java: `Collections.sort`