

# Sorting

October 20, 2006

ComS 207: Programming I (in Java)  
Iowa State University, FALL 2006  
Instructor: Alexander Stoytchev

© 2004 Pearson Addison-Wesley. All rights reserved

## Midterm 2

- Next Tuesday (Oct 24, 2006)
- Location: Hoover Hall, room 2055
- Time: 6:30pm – 7:45pm
- Try to be there at least 10 minutes early.
- If you need extra time you can have it but this is **\*\*NOT\*\*** a 3 hour exam!

© 2004 Pearson Addison-Wesley. All rights reserved

## Midterm Format

- Covers Sections 1-5 & 7
- Also, Searching and Sorting
- Format same as Midterm 1

© 2004 Pearson Addison-Wesley. All rights reserved

## Midterm Format (last semester)

- True/False (10 x 1p each = 10p)
- Short answer (5 x 2p each = 10p)
- Code Snippets (5 x 3p each = 15p)
- Other Stuff (3 x 5p each = 15p)
- Other Stuff (3 x 5p each = 15p)
- Program 1 (15p)
- Program 2 (15p)
- Program 3 (15p)
- Program 4 (20p)
  
- TOTAL (130p)

© 2004 Pearson Addison-Wesley. All rights reserved

## Midterm Format

- You don't need to get all 130 points to get an A
- 100 is a 100
  
- You must get at least 65 points in order to pass this exam

© 2004 Pearson Addison-Wesley. All rights reserved

## Midterm Format

- Drop Deadline is next Friday (Oct 27)
  
- I cannot guarantee that all exams will be graded by then.
  
- If you believe that you did not do well please ask me to grade your exam first at the time when you are submitting it.

© 2004 Pearson Addison-Wesley. All rights reserved

## Quick review of last lecture

© 2004 Pearson Addison-Wesley. All rights reserved

## Binary Search

- At each step it splits the remaining array elements into two groups
- Therefore, it is faster than the linear search
- Works only on an already SORTED array
- Thus, there is a performance penalty for sorting the array

© 2004 Pearson Addison-Wesley. All rights reserved [http://web.ics.purdue.edu/~cs154/lectures/lecture011.htm]

### Example: Successful Binary Search

© 2004 Pearson Addison-Wesley. All rights reserved [http://web.ics.purdue.edu/~cs154/lectures/lecture011.htm]

### Example: BinarySearch.java

© 2004 Pearson Addison-Wesley. All rights reserved

```

while (first <= last) {
    if (a[mid] == target)
        return mid;
    else if (a[mid] > target)
        last = mid - 1;
    else
        first = mid + 1;
}
return -1;

```

© 2004 Pearson Addison-Wesley. All rights reserved

### Analysis of Searching Methods

- For an array of size  $n$
- Sequential Search (Average-Case)  $n/2$
- Sequential Search (Worst-Case)  $n$
- Binary Search (Average-Case)  $\log(n)/2$
- Binary Search (Worst-Case)  $\log(n)$

© 2004 Pearson Addison-Wesley. All rights reserved

Sorting  
Not in the Textbook

PEARSON Addison-Wesley  
© 2004 Pearson Addison-Wesley. All rights reserved.

### Insertion Sort

© 2004 Pearson Addison-Wesley. All rights reserved. [http://web.ics.purdue.edu/~cs154/lectures/lecture010.htm]

### Example: Insertion Sort

© 2004 Pearson Addison-Wesley. All rights reserved. [http://web.ics.purdue.edu/~cs154/lectures/lecture010.htm]

### Animations for Insertion Sort

Number of Comparisons: 242  
Number of Swaps: 241

© 2004 Pearson Addison-Wesley. All rights reserved. [http://maven.smith.edu/~thiebaut/java/sort/demo.html]

### Animations of Sorting Algorithms

- <http://maven.smith.edu/~thiebaut/java/sort/demo.html>
- <http://www.cs.ubc.ca/spider/harrison/Java/sorting-demo.html>

© 2004 Pearson Addison-Wesley. All rights reserved.

### Swapping Array Elements

Assume smallest element has been found at  $x[4]$   
Want to swap with  $x[0]$  element, bring smallest to beginning

Three steps

© 2004 Pearson Addison-Wesley. All rights reserved. [http://www.scs.ryerson.ca/jpanar/w2006/cps125/example.gifs/ch7/Arr.Swap.gif]

## Java code

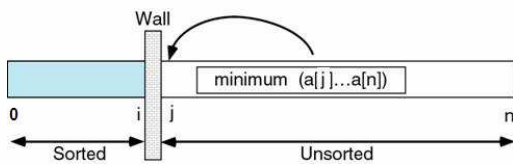
```
// Swap a[i] with the smallest element  
  
int temp = a[i];  
a[i] = a[minIndex];  
a[minIndex] = temp;
```

© 2004 Pearson Addison-Wesley. All rights reserved

## Example: InsertionSort.java

© 2004 Pearson Addison-Wesley. All rights reserved

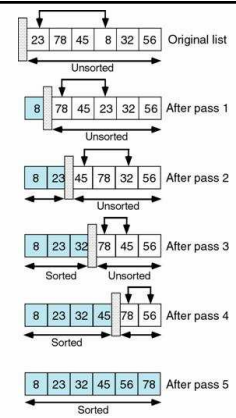
## Selection Sort



© 2004 Pearson Addison-Wesley. All rights reserved

[<http://web.ics.purdue.edu/~cs154/lectures/lecture010.htm>]

## Example: Selection Sort



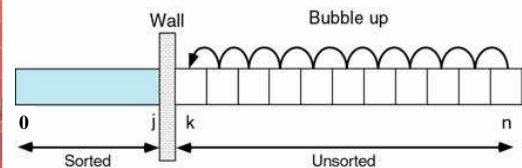
© 2004 Pearson Addison-Wesley. All rights reserved

[<http://web.ics.purdue.edu/~cs154/lectures/lecture010.htm>]

## Example: SelectionSort.java

© 2004 Pearson Addison-Wesley. All rights reserved

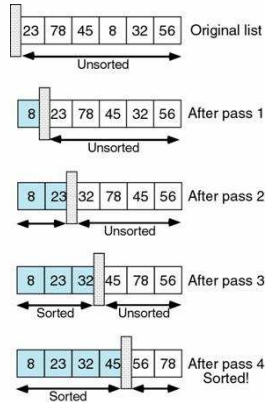
## Bubble Sort



© 2004 Pearson Addison-Wesley. All rights reserved

[<http://web.ics.purdue.edu/~cs154/lectures/lecture010.htm>]

## Example: Bubble Sort



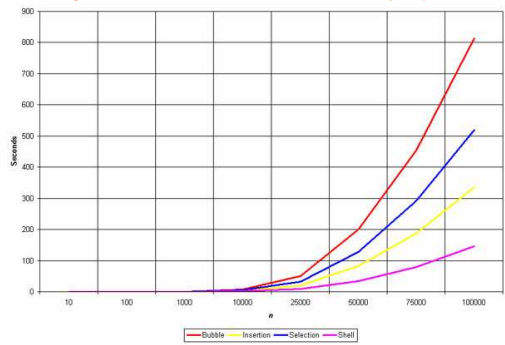
© 2004 Pearson Addison-Wesley. All rights reserved

[<http://web.ics.purdue.edu/~cs154/lectures/lecture010.htm>]

## Example: BubbleSort.java

© 2004 Pearson Addison-Wesley. All rights reserved

## Analysis: all three run in $O(n^2)$ time



© 2004 Pearson Addison-Wesley. All rights reserved

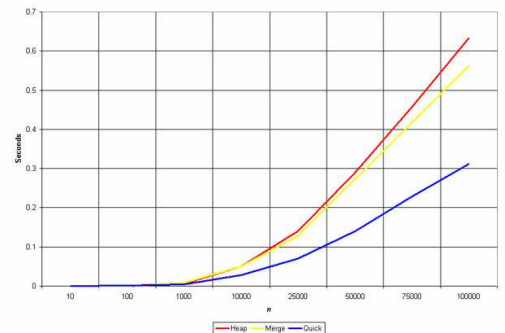
[<http://linux.wku.edu/~lmonml/algorsort/sort.html>]

## Analysis

- There are faster sorting algorithms
  - Heap sort
  - Quick sort
  - Merge Sort
- We will not cover those but feel free to study them on your own.
- They run in  $O(n \log n)$  time.

© 2004 Pearson Addison-Wesley. All rights reserved

## $O(n \log n)$ sorting algorithms



© 2004 Pearson Addison-Wesley. All rights reserved

[<http://linux.wku.edu/~lmonml/algorsort/sort.html>]

THE END

© 2004 Pearson Addison-Wesley. All rights reserved