Sorting

October 19, 2007

Midterm 2

• Next Tuesday (Oct 23, 2007)
• Location: Curtiss Hall, room 127 (classroom)
• 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!

Midterm Format

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

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)

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

Midterm Format

• Drop Deadline is next Friday (Oct 26)
• 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.
Quick review of last lecture

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

Example: Successful Binary Search

Example: BinarySearch.java

Analysis of Searching Methods
- For an array of size $n$
  - Sequential Search (Average-Case) $\frac{n}{2}$
  - Sequential Search (Worst-Case) $n$
  - Binary Search (Average-Case) $\frac{\log(n)}{2}$
  - Binary Search (Worst-Case) $\log(n)$
Sorting

Not in the Textbook

Example: Insertion Sort

Animations for Insertion Sort

Animations of Sorting Algorithms

- http://maven.smith.edu/~thiebaut/java/sort/demo.html


Swapping Array Elements

Assume the selected element has been found at \( a[p] \).

Want to swap with \( a[q] \) element, bring selected to beginning.

Three steps:

1. \[ a[p] \leftarrow \text{temp} \]
2. \[ \text{temp} \leftarrow a[q] \]
3. \[ a[q] \leftarrow a[p] \]
Java code

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

Selection Sort

Example: SelectionSort.java

Bubble Sort

Example: SelectionSort.java
Example: Bubble Sort

BubbleSort.java

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

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.

$O(n \log n)$ sorting algorithms

THE END