Searching, Memory, and Bugs

11-8-2010

Opening Discussion

- IcP solutions.
- Do you have any questions about the assignment?
- PAD2 scheduling
- Minute essay comments:
  - How do you pick which sort to use?
  - ETAs on chapters.

Binary Search

- If the data is sorted, we can do something much better.
- We check the middle to see if it matches. If it does, return it. Otherwise, see if what we want is above or below the middle and repeat the process on only that half.
- This continually divides the things we are searching in half.
- Order?

Performance of Binary Search

- Dividing something by a fixed fraction repeatedly leads to O(log n) speed.
- O(log n) is much better than O(n) when n is large. To see this, consider a base 2 log for 1000, 1000000, or 1000000000.

Computer Memory

- The memory that a program uses is broken into two different parts.
  - Stack – This holds local variables. Every function/method call gets a new “frame” on the stack. Efficient, but limited.
  - Heap – All objects in Scala are allocated on the heap. It is big and flexible, but disorganized.
- Other languages allow you more direct control over memory. This has the potential to lead to errors.

Classification of Bugs

- We classify the errors that occur in programs in three broad groups.
  - Compile Errors – Found by the compiler. Gets a reasonable error message and line number.
  - Runtime Errors – Program crashes while running for a particular input. Gives type of error and line.
  - Logic Errors – Code runs fine, but does wrong thing. No information given to help you.
- You want to have your errors be higher up on this list because it gives you more information and makes it easier to fix.

Minute Essay

- Do you have any suggestions for assignments or examples?
- Interclass problem:
  - Write a binary search that will run through the weather data to find a particular month from a given year.