

Sorting and Searching

3-28-2011

Opening Discussion

- Minute essay comments:
 - Quiz content. Files, case classes, GUIs (No graphics)
 - Sorts in the API.
- IcP solutions

Watching Them Work

- One advantage of doing graphics before sorting is that we can write code to visualize what is happening when we sort numbers with these sorts.

Insertion Sort

- Inner loop:
 - Take the next element and shift it down to the right spot.
 - Use a while loop because we don't know how far it will go.
- Outer loop:
 - Run through all the elements starting with the second.
- This sort is actually a bit faster (factor of 2) on random data. It is really efficient on nearly sorted data.

Searching

- Many times we have to search in our data for where something is.
- If the data is not sorted, we have to use a linear search which will look at every element, one after another, to see if any matches what we want.
- This is $O(n)$.
- The methods on collections in Scala use this approach.

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 10000000000.

Minute Essay

- What questions do you have?
- Keep CSCI 1321 in mind when building your schedule for the fall. My section, which is the only one using Scala, will meet 11:30-12:20 MWF.