Faster Sort Algorithms

10-12-2001

Opening Discussion

- What did we talk about last class?
- Most of you were able to trace a sort function properly though some of you said you traced a min sort when you actually traced an insertion sort.
- Today's material is advanced and therefore somewhat optional. I will field any questions you want to ask for as long as you ask them (or look at two more sorts).

Divide and Conquer

- One of the standard methods of approaching problems in CS is the divide and conquer method. This is where a problem is first split into smaller parts and then each of the parts is tackled by itself.
- We saw this yesterday in the binary search. Today we will look at two sort methods that do this as well.

Merge Sort

- The idea behind merge sort is to break the array into two pieces and then sort each one separately then "merge" then back together.
- We recursively break the array down until we get to single elements. We merge those back together in proper order as we "pop" back up the recursion stack.

Overview of Merge Sort

- Checks if there is only one element and if so just return.
- Recurses on two halves and assumes those halves are properly sorted when they return.
- Merges the two halves together by running down the arrays. And returns.
- Recurses down log(n) times and does O(n) work each time.

Quicksort

- The problem with merge sort is that it can't merge efficiently in a single array. It needs a second array to work with. Quicksort corrects that.
- Quicksort picks a "pivot" element and does one pass through the array making sure all elements are on the "right" side of the pivot (depending on whether they are greater or less than it).

More Quicksort

- It then calls itself recursively on the part of the array below the pivot and on the part above it.
- The recursion terminates when it gets called with only one element.
- Quicksort does O(n) work for each call. With good pivots quicksort recurses log(n) times. However, it has a worst case performance where it recurses O(n) times.

Minute Essay

- Today's methods are "faster" than those we discussed last time. Compare n^2 to n*log(n) for n values of 10, 1000, and 1000000. Assume a base 10 log (base 2 is more accurate but the difference is just a factor of a bit over 3).
- Assignment #3 is due Monday when we will discuss strings and multidimensional arrays.
- I'll be giving a talk on planet formation tomorrow at 4pm in the science lecture hall.