Sorting

11-19-2003

Opening Discussion

- What did we talk about last class?
- Do you have any questions about the assignment?
- Assume I give you a list of words and tell you to put them in alphabetical order. How would you go about doing that? What would an algorithm look like to do that?

Motivation

- It is very often convenient to organize data into some form of logical ordering. This helps humans viewing it, but can also greatly speed processing of the data on a computer.
- The process of putting data in this order is called sorting and that is our topic today. It is something you can all do, but describing a pedantic safe algorithm can be different.

The Basic Idea

- There are many different ways to sort data. They all have basic features in common though. We have to compare elements of our collection and reorder then so that they are in the order we want.
- The question is, how do we pick which ones to compare and when do we move things around?

Bubble Sort

- The simplest sort is called the bubble sort. The name comes from the fact that items "bubble" through to one end of the array.
- For a bubble sort we have two loops. The inner loop goes through the elements of the array comparing adjacent items. If they are out of order they get swapped. The outer loop makes sure this keeps happening until everything is in order.

Selection Sort

- Bubble sort is extremely inefficient. We can do better with a selection sort (also called a min or max sort).
- Again we have two loops. The inner one "looks" for the minimum or maximum element of the unsorted part of the array. Once found, that element is moved into place. The outer loop makes sure that this process is done for all the elements of the array.

Order Analysis (in brief)

- In CS we worry a lot about how much work a computer does to complete a given task. Formal analysis of algorithms allows us to say which are good and which are bad.
- These sorting algorithms are $O(n^2)$. This means that the "amount of work" done to sort n elements increases the same way the function n^2 does.

A Closer Look

- For a proper order analysis we have to tell what type of operations we are counting.
- If we are counting comparisons, both algorithms are $O(n^2)$. However, if we are counting memory moves, then selection sort gets a big upper hand. This is because its swap is only inside one loop and is thus O(n). Selection sort is much better for large structures.

Code

- Let's write the code for bubble sort and selection sort to see how they look and behave.

Minute Essay

- Take the array {5,8,2,7,3} and show me what it loops like after each pass through a bubble sort and a selection sort. Remember that selection sort swaps the element it finds.
- The assignment is due next Monday.