Sorting with Structs

11-21-2003

Opening Discussion

- What did we talk about last class?
- Do you have any questions about the assignment?

Quick Review

- Sorting is a very important activity on computers for many reasons.
- Last time we talked about two simple sorts.
- Bubble Sort - compares adjacent elements and swaps them if they are out of order.
- Selection Sort - finds the next smallest or greatest element and swaps it into place.

Insertion Sort

❚ A third "simple" sort is what is called insertion sort. With this sort, we slowly build up a sorted part of the array at the front by inserting each element where it goes.

❚ This type of sort doesn't use swaps. Instead, it keeps a temporary copy of the element being inserted and moves things over by copying over other elements.

Strengths and Weaknesses

❚ Bubble sort is easy to write. A flagged bubble sort can do well with nearly sorted data.

❚ Selection sort always does almost exactly the same amount of work. It does $O(n)$ copies which can help with big structures.

❚ Insertion sort is ideal for nearly sorted data. It can do less then $O(n^2)$ compares and copies with nearly sorted data.

Sorting Complex Data

❚ The sorting algorithms that we discussed are used for any type of data. However, the code we wrote was for just a basic array. The logic gets more complex when the data has more pieces.

❚ Go back to the idea of having arrays of names and grades and wanting to sort the students in alphabetical order. What do we need to change in the code to make that work?

Sorting with Structures

- If the array that we have is one of structures, our job is a bit less complex because the copying of the structures can be done fairly easily.
- It should be noted that copying structure in C does what is called a "shallow" copy so structures with pointers can behave in ways other than what you were expecting. This isn't a huge problem for sorts.

Code

- Let's write some code to sort an array of structures instead of just an array of ints so that we can see how the code is altered. What types of flexibility do we lack with what we are doing here?

Minute Essay

- Show me the resulting array after each pass through an insertion sort with the following array: {6,8,2,5,3}.