

ECE 222 System Programming Concepts

Lecture notes – Introduction to Sorting

The problem of sorting can have two meanings. In one context, it refers to grouping similar items; for example, sorting laundry into piles of shirts and socks. In another context, it refers to ordering items according to some rules; for example, organizing names alphabetically in a phone book. We are concerned with the second problem, that of ordering.

Sorting (ordering) has a long history in computing. A large number of methods have been developed to sort data. In this lecture, our goal is to introduce the topic by going through one of the simplest sorting methods known: selection sort.

Assume we have an unordered list of data:

7 4 8 3 5

Assume we want to reorder it, largest to smallest:

8 7 5 4 3

Selection sort works as follows:

Loop i = index of 1st element to 2nd-to-last element
 Find the largest element from i to end of list
 Swap largest with value at index i

On our example data, i starts at the beginning, and we find the largest:

i		largest		
7	4	8	3	5

After the swap, we have:

8 4 7 3 5

Now i moves forward, and we find the largest in the remaining data:

	i	largest		
8	4	7	3	5

After the swap, we have:

8 7 4 3 5

Continuing in this manner, we see the following steps:

		i		largest	
8	7	4	3	5	
8	7	5	3	4	after swap
			i	largest	
8	7	5	3	4	
8	7	5	4	3	after swap

Note that we stop here, with i at the 2nd-to-last index, because there is nothing to do on the last index; it is by default already in the correct order.

How do we implement this in C?

[show sort1.c example, and go through it]

Selection sort is important because of how easy it is to understand, and how easy it is to implement. On relatively small data sets, where computing power is not an obstacle, and the time to compute the result is meaninglessly small or unimportant, this method is often to be preferred. Its simplicity of implementation helps avoid implementation errors.

The formal study of sorting is reserved for a course in data structures or algorithms, where numerous variations on sorting are studied. At Clemson, this happens in ECE 223.

Now that you understand how selection sort works, complete the in-class exercise sort2.c. It has some compile-time and run-time errors in it. The program should sort strings in alphabetical order.

[sort2.c – in-class problem]