

COP 3502 Suggested Program Edits: Sorting (Week 9 Programs)

1) Edit Bubble Sort so that if a whole iteration completes without making any swaps, then you don't run any more iterations and return.

2) Rewrite insertion sort so that the sorted list is built from the right side and not the left. So, for example, if the array first has:

3 2 8 1 6

It would look like this after each iteration:

3	2	8	1	6	(inserted 1 into list with 6)
3	2	1	6	8	(inserted 8 into list 1,6)
3	1	2	6	8	(inserted 2 into list 1,6,8)
1	2	3	6	8	(inserted 3 into list 1,2,6,8)

3) In selection sort the way it was taught in class, we place the maximum element of the array at the end first. Reverse the logic so that the first item that is selected is the smallest, the second item selected is the second smallest, etc.

4) Rewrite merge sort so that it splits the array into the left size that is one third (roughly) the size of the full array and the right side so that it is two thirds the size of the fully array. Run this version against the usual version and time the code on large arrays. Is this version faster or slower? Why?

5) Edit the posted file with quick sort and try a base case of various sizes ranging from 5 to 100, where in the base case, you sort the subarray using insertion sort. Time the various versions on large arrays. Which one works the best?