Mark McKelvy
Data Structures 3:00 MWF
1/23/03
Pg. 26 1-3,5-6,8-9

1. Which of the sorting methods studied in this section allows a possible early exit from its inner loop? Bubble sort What is the potential advantage of using this early exit? The sort saves time, and frees up system resources sooner
2. Which of the sorting methods in this section allows a possible early exit from its outer loop? Selection sort What is the potential advantage of using this early exit? This sort, which is less system intensive, will finish earlier
3. Which of the sorting methods in this section does not allow for the possibility of an early exit from its inner and outer loops? Insertion sort What potential advantage does this method have over the other two methods that were presented? This method takes advantage of the partial ordering of a list
5.

Selection Sort:
The list (before sorting) is as follows:
list[0] = 43
list[1] = 40
list[2] = 18
list[3] = 24
list[4] = 39
list[5] = 60
list[6] = 12

Pass 1 looks like this:
list[0] = 12
list[1] = 40
list[2] = 18
list[3] = 24
list[4] = 39
list[5] = 60
list[6] = 43

Pass 2 looks like this:
list[0] = 12
list[1] = 18
list[2] = 40
list[3] = 24
list[4] = 39
list[5] = 60
list[6] = 43

Pass 3 looks like this:
list[0] = 12
list[1] = 18
list[2] = 24
list[3] = 40
list[4] = 39
list[5] = 60
list[6] = 43

Pass 4 looks like this:
list[0] = 12
list[1] = 18
list[2] = 24
list[3] = 39
list[4] = 40
list[5] = 60
list[6] = 43

Pass 5 looks like this:
list[0] = 12
list[1] = 18
list[2] = 24
list[3] = 39
list[4] = 40
list[5] = 60
list[6] = 43

Pass 6 looks like this:
list[0] = 12
list[1] = 18
list[2] = 24
list[3] = 39
list[4] = 40
list[5] = 43
list[6] = 60

6.

Insertion Sort:
The list (before sorting) is as follows:
list[0] = 43
list[1] = 40
list[2] = 18
list[3] = 24
list[4] = 39
list[5] = 60
list[6] = 12

Pass 1 looks like this:
list[0] = 40
list[1] = 43
list[2] = 18
list[3] = 24
list[4] = 39
list[5] = 60
list[6] = 12

Pass 2 looks like this:
list[0] = 18
list[1] = 40
list[2] = 43
list[3] = 24
list[4] = 39
list[5] = 60
list[6] = 12

Pass 3 looks like this:
list[0] = 18
list[1] = 24
list[2] = 40
list[3] = 43
list[4] = 39
list[5] = 60
list[6] = 12

Pass 4 looks like this:
list[0] = 18
list[1] = 24
list[2] = 39
list[3] = 40
list[4] = 43
list[5] = 60
list[6] = 12

Pass 5 looks like this:
list[0] = 18
list[1] = 24
list[2] = 39
list[3] = 40
list[4] = 43
list[5] = 60
list[6] = 12

Pass 6 looks like this:
list[0] = 12
list[1] = 18
list[2] = 24
list[3] = 39
list[4] = 40
list[5] = 43
list[6] = 60

8. This sort algorithm resembles the selection sort. The difference is where the int's are declared in the code. A tracing of this type of algorithm can be found for question #5.
9. This sort algorithm most closely resembles the insertion sort. The variable names are different, but the method is still the same. This type of algorithm was traced in question #6.