

Practical Exercise 11 – Sorting

Part 1 – Discussion

1. Given the following array:

44, 78, 22, 7, 38

Assume that the items in the array need to be sorted into ascending order. Step through the sort process manually and show the contents of the array at each step of the sort. Use and compare the following sort methods:

i. Bubble sort

After first iteration:

44, 22, 7, 38, 78 (Bubble the largest element to the last position)

After 2nd iteration:

22, 7, 38, 44, 78 (Bubble the 2nd largest element to the 2nd last position)

After 3rd iteration:

7, 22, 38, 44, 78 (Bubble the 3rd largest element to the 3rd last position)

Finished.

ii. Selection sort

Initial:

44, 78, 22, 7, 38

After first iteration:

44, 38, 22, 7, 78 (Swap the largest element to the last position)

After 2nd iteration:

7, 38, 22, 44, 78 (Swap the 2nd largest element to the 2nd last position)

After 3rd iteration:

7, 22, 38, 44, 78 (Swap the 3rd largest element to the 3rd last position)

Finished.

iii. Insertion sort

Initial:

44, 78, 22, 7, 38

After first iteration:

22, 48, 78, 7, 38

(Find the first pair which is out of order (which is (78, 22), then swap them, then sink the swapped element (22) to the lowest logical position)

After 2nd iteration:

7, 22, 48, 78, 38

(Find the first pair which is out of order (which is (78, 7), then swap them, then sink the swapped element (7) to the lowest logical position)

After 3rd iteration:

7, 22, 38, 44, 78

(Find the first pair which is out of order (which is (78, 38), then swap them, then sink the swapped element (38) to the lowest logical position)

Finished.

iv. Merge sort

v. Quick sort

2. Show the heaps when inserting each of these keys in this order: 44, 66, 33, 88, 77, 55, 22. Also, show the heaps when retrieving each of the keys.
3. Discuss the efficiency of each of the sorting methods listed above in terms of its complexity by using the Big-O notation.

Part 2 – Programming Exercise

Question 1

Define a test program that puts 20 random integers between 0 and 100 into an array. The program should print out a random array of integers of size 20.

[Note that in Java, there is a method `Math.random()`, which returns a double value between 0.0 and 1.0. And there is another method `Random.nextInt(int n)`, which returns a random value in the range of 0 (inclusive) and n (exclusive).]

Question 2

In addition to the test program you have constructed for Question 1, write the program segment which sorts the random array of integers into increasing order of integers using *quick sort* or *merge sort*. The program should print out each of the comparison and swapping routine until the array is sorted.