CSSE 220 – Object-Oriented Software Development
Rose-Hulman Institute of Technology

Worksheet 19

Name (Print):______________________________ **Section:**______________

1. A sorting algorithm is a systematic procedure for ______________________
2. ___________ decides the new order of elements
3. Selection Sort has 2 parts: 1) ___________ part at the beginning 2) ___________ part is the rest of array
4. Suppose the selection sort algorithm from class is applied to the initial array:

9	5	7	10	18	1	12	8	16	4

In the boxes below, show the state of the array immediately following each of the first two iterations. Clearly mark the sorted part (with a vertical line) and the unsorted part of the array after each iteration for 1st, 2nd iterations. (Initially the sorted part of the array is empty.)

0th iteration:

9	5	7	10	18	1	12	8	16	4

1st iteration:

2nd iteration:

5. Define the profiling: __________________________
6. Complete the profiling analysis for SelectionSort.java and update the excel table (see link in the course schedule).

Size, n	**Run-time $t(n)$ (ms)**	**Size, n**	**Run-time $t(n)$ (ms)**
10 000	__________	60 000	__________
20 000	__________	70 000	__________
30 000	__________	80 000	__________
40 000	__________	90 000	__________
50 000	__________	100 000	__________

7. Select which algorithm does Selection Sort folow (is similar/behaves like):
   1) O(N) 2) O(N^2) 3) None

8. Suppose you have 10 people in the array and they all do handshakes. How many handshakes are in total? ________________

9. Review the code SelectionSort:
   1). In selection sort, for an array of length n, how many times does the OUTER LOOP execute, in terms of n? ________________

   2). In selection sort, for an array of length n, how many times is compareTo() called during the FIRST iteration of the outer loop? ________________

   3). ... during the SECOND iteration of the outer loop? ________________

   4). ... during the SECOND-TO-LAST iteration of the outer loop? ________________

   5). ... during the LAST iteration of the outer loop? ________________

10. Write a formula solution in n for the number of times selectionSort() calls compareTo() for an array of size n:

    ____

11. What is runtime in terms of Big-O?

```
public static int countOccurences(int value, int[] array) {
            int count = 0;
            int i = 0;
            while(true) {
                    if(value == array[i])
                          ++count;
                    if(i == array.length / 2)
                          return count;
                    i++;
            }
    }
//Runtime: ________________________
```

12. Summary Table for Selection Sort:

Case	Number of Comparisons	BigO
Worst Average Best		

13. Suppose the insertion sort algorithm is applied to the initial array above. Show the state of the array immediately following each of the first three iterations of the outer loop. Clearly mark (as in 0th is alreday marked for you) the sorted part of the array after each iteration.

0^{st} iteration:

9 ‖	5	7	10	18	1	12	8	16	4

1^{st} iteration:

2^{nd} iteration:

3^{rd} iteration:

14. Complete the profiling analysis for InsertionSort.java and update the excel table (see link in the course schedule).

Size, n	**Run-time $t(n)$ (ms)**	**Size, n**	**Run-time $t(n)$ (ms)**
10 000	______________	60 000	______________
20 000	______________	70 000	______________
30 000	______________	80 000	______________
40 000	______________	90 000	______________
50 000	______________	100 000	______________

15. Summary Table for InsertionSort:

Case	BigO
Worst (reversed)	
Average	
Best (sorted)	

16. Summary Table for Binary Search:

Case	BigO
Worst	
Best	
(ony one comparison is needed)	

17. What is run-time in terms of Big-O?

```java
public static void function3(int[] array) {
     for(int i = 1; i <= array.length; i++) {
         for(int j = array.length; j >= 1; j = j / 2) {
              if(array[i-1] <= array[j-1]) {
                  array[j-1] = array[i-1];
              }
          }
       }
}
//Runtime: ________________________

```

18. Summary Table for MergeSort:

Case	BigO
Worst Best	

19. Suppose the merge sort algorithm from class is applied to the initial array.

9	5	7	10	18	1	12	8	16	4

Show the state of the two sub-arrays immediately before the final merge.

