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

## Worksheet 21

Name (Print):_____ Section:_____

1. Express Cost in terms of BigO for ArrayList:

| Operation | Cost |
|---|---|
| get(index)<br>add(index, element)<br>remove(index)<br>size() | |

2. A sorting algorithm operates by _____ the list, _____ elements, and _____ them _____

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. Comparison Math Formula for selection algorithm:

_____

6. Summary Table for Selection Sort:

| Case | BigO |
|---|---|
| Worst<br>Best | |

7. What is Big-O?

```
1  public static int countOccurences(int value, int[] array) {
2              int count = 0;
3              int i = 0;
4              while(true) {
5                      if(value == array[i])
6                             ++count;
7                      if(i == array.length / 2)
8                             return count;
9                      i++;
10             }
11     }
12 //BigO: _________________________
```

8. 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:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|

9. Summary Table for InsertionSort:

| Case | BigO |
|---|---|
| Worst (reversed)<br>Average<br>Best (sorted) | |

10. True/False Binary search must be sorted

11. Summary Table for Binary Search:

| Case | BigO |
|---|---|
| Worst<br>Best | |