Name: ____________________

UserID: ____________________

Lab Section: ____________________

Date: ____________________

COMP 2210
Exam 1 Sample, Version: $\boxed{1}$

Section 1. Sorting

1. Which sorting algorithm has the following time complexity profile? Best case: $O(N \log N)$, Average case: $O(N \log N)$, Worst case: $O(N^2)$.
   (a) selection sort
   (b) insertion sort
   (c) merge sort
   (d) quicksort

2. Suppose an array with its elements already in sorted (i.e., ascending) order is passed to a sorting method and you observe that the method takes time proportional to N. Which sorting algorithm is implemented in this method?
   (a) selection sort
   (b) insertion sort
   (c) merge sort
   (d) quicksort

3. Suppose an implementation of quicksort chooses the pivot as the left-most element in each partition of the array. Which of the following arrays would be most likely to evoke a worst case running time?
   (a) `[2, 4, 6, 8, 10, 12]`
   (b) `[6, 2, 8, 10, 4, 12]`
   (c) `[10, 2, 6, 4, 12, 8]`
   (d) `[8, 4, 2, 10, 6, 12]`

4. Which of the sorting algorithms, as implemented in class, was not a *stable* sort?
   (a) selection sort
   (b) insertion sort
   (c) merge sort
   (d) quicksort

5. Given the array `a = [5, 9, 1, 6, 2, 3, 4, 7, 8]`, which sorting algorithm would perform the following sequence of array modifications?

```
[5, 9, 1, 6, 2, 3, 4, 7, 8]
[1, 5, 9, 6, 2, 3, 4, 7, 8]
[1, 5, 6, 9, 2, 3, 4, 7, 8]
[1, 2, 5, 6, 9, 3, 4, 7, 8]
[1, 2, 3, 5, 6, 9, 4, 7, 8]
[1, 2, 3, 4, 5, 6, 9, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 9, 8]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

(a) selection sort
(b) insertion sort
(c) merge sort
(d) quicksort

6. Which of the sorting algorithms, as implemented in class, was not an *in-place* sort?

(a) selection sort
(b) insertion sort
(c) merge sort
(d) quicksort

7. Suppose an array with its elements in reverse (i.e., descending) order is passed to a sorting method and you observe that the method takes time proportional to N^2. Which sorting algorithm is definitely *not* implemented in this method?

(a) selection sort
(b) insertion sort
(c) merge sort
(d) quicksort

8. Suppose we have a class named Point that represents ordered pairs of numeric data (x, y). Also suppose that the *natural order* of Point is defined only on the x field. Thus, $(4, 5) < (5, 9)$ and $(4, 5) = (4, 9)$. Given the array of Point objects

```
a = [(5,1), (4,2), (3,3), (5,4), (2,5), (1,6), (5,7)]
```

if the method call `sort(a)` arranges the array to be

```
a = [(1,6), (2,5), (3,3), (4,2), (5,1), (5,7), (5,4)]
```

what can you conclude about the sorting algorithm that this method implements?

(a) The sorting algorithm is adaptive.
(b) The sorting algorithm is not adaptive.
(c) The sorting algorithm is stable.
(d) The sorting algorithm is not stable.

9. Which sorting algorithm has the following time complexity profile? Best case: $O(N^2)$, Average case: $O(N^2)$, Worst case: $O(N^2)$.

(a) selection sort
(b) insertion sort
(c) merge sort
(d) quicksort

10. Which of the arrays below could be the result of partitioning `a = [9, 3, 4, 1, 6, 8, 5, 2, 7]` using 4 as the pivot, as discussed in the quicksort implementation in class?

    (a) [9, 8, 7, 6, 5, **4**, 3, 2, 1]

    (b) [3, 1, 2, 6, 8, 5, 7, 9, **4**]

    (c) [**4**, 1, 2, 3, 5, 6, 7, 8, 9]

    (d) [3, 1, 2, **4**, 6, 8, 5, 7, 9]

11. Given the array `a = [5, 6, 3, 1, 8, 9, 4, 2, 7]`, which sorting algorithm would perform the following sequence of array modifications?

```
[1, 6, 3, 5, 8, 9, 4, 2, 7]
[1, 2, 3, 5, 8, 9, 4, 6, 7]
[1, 2, 3, 5, 8, 9, 4, 6, 7]
[1, 2, 3, 4, 8, 9, 5, 6, 7]
[1, 2, 3, 4, 5, 9, 8, 6, 7]
[1, 2, 3, 4, 5, 6, 8, 9, 7]
[1, 2, 3, 4, 5, 6, 7, 9, 8]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

    (a) selection sort

    (b) insertion sort

    (c) merge sort

    (d) quicksort

12. Which sorting algorithm has the following time complexity profile? Best case: $O(N \log N)$, Average case: $O(N \log N)$, Worst case: $O(N \log N)$.

    (a) selection sort

    (b) insertion sort

    (c) merge sort

    (d) quicksort

13. Which sorting algorithm has the following time complexity profile? Best case: $O(N)$, Average case: $O(N^2)$, Worst case: $O(N^2)$.

    (a) selection sort

    (b) insertion sort

    (c) merge sort

    (d) quicksort

Section 2. Efficiency and Algorithm Analysis

14. Which big-oh expression best characterizes the worst case time complexity of the following method?

```
int count = 0;
for (int i = 1; i < N; i++) {
    int j = 1;
    while (j < N) {
        j++;
        count++;
    }
}
```

(a) $O(1)$

(b) $O(\log N)$

(c) $O(N)$

(d) $O(N^2)$

15. Consider applying binary search on an array of N elements. After the k^{th} element has been examined, what is the size of the remainder of that array that still needs to be searched?

(a) $N/2k$

(b) $N/2^k$

(c) $\log N$

(d) $\log k$

16. Suppose you attempted to empirically discover the big-oh running time of a program, and you were able to generate the following timing data.

N	Time	Ratio
250	0.061	N/A
500	0.042	1.35
1000	0.112	2.67
2000	0.340	3.04
4000	1.298	3.82
8000	5.334	4.11
16000	21.881	4.10
32000	85.763	3.92
64000	345.634	4.03

In the table above, the N column records the size of the input for each run, the Time column records the elapsed time in seconds for each run, and Ratio is the elapsed time for the current run divided by the elapsed time for the previous run (i.e., $Time_{\text{i}}/Time_{\text{i-1}}$).
Based on the timing data presented in this table, what is the most reasonable conclusion regarding the underlying big-oh time complexity of the program being timed?

(a) $O(\log N)$

(b) $O(N)$

(c) $O(N^2)$

(d) $O(N^4)$

17. Which big-oh expression best characterizes the worst case time complexity of the following method?

```
count = 0;
int i = 1;
while (i < N) {
    count++;
    i = i * 2;
}
```

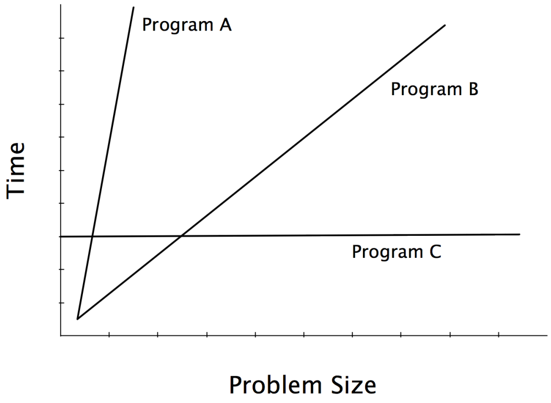
(a) $O(1)$

(b) $O(\log N)$

(c) $O(N)$

(d) $O(N^2)$

18. The graph below depicts the growth rates of three programs that solve the same problem. Which program does the graph suggest would offer the best performance as the problem size scales up?



(a) Program A

(b) Program B

(c) Program C

(d) There is not enough information to decide.

19. Which big-oh expression best characterizes the worst case time complexity of the following method?

```
count = 0;
int i = N;
while (i > 1) {
    count++;
    i = i - 2;
}
i = N;
while (i > 1) {
    count++;
    i = i / 2;
}
```

(a) $O(N \log N)$

(b) $O(N^2)$

(c) $O(\log N)$

(d) $O(N)$

20. Which big-oh expression best characterizes the worst case time complexity of the following method?

```
count = 0;
for (int i = 1; i < 10000; i++) {
    for (int j = 1; j < 500; j++) {
        for (int k = 1; k < 10000; k++) {
            count++;
        }
    }
}
```

(a) $O(1)$

(b) $O(\log N)$

(c) $O(N)$

(d) $O(N^2)$

21. As a function of N, how many times is the method `foo` (line 3) called in the code below?

```
1 for (int j = 0; j < N; j++) {
2     for (int k = j; k < N; k++) {
3         foo();
4      }
5 }
```

(a) $2N + 1$

(b) $2N^2$

(c) $(N^2 + N)/2$

(d) $(N^2 - N + 1)/2$

22. Which big-oh expression best characterizes the worst case time complexity of the following method?

```
int count = 0;
for (int i = 1; i <= N; i++) {
    int j = N;
    while (j > 1) {
        for (int k = 1; k < N; k = k + 2) {
            count++;
        }
        j = j / 2;
    }
}
```

(a) $O(N^3)$

(b) $O(N^2)$

(c) $O(N^2 \log N)$

(d) $O(N \log \log N)$

23. Which list below orders the functions from slowest growth rate to fastest growth rate?

(a) $\log N$, N, $N \log N$, N^2

(b) N^2, $N \log N$, N, $\log N$

(c) N, N^2, $N \log N$, $\log N$

(d) $\log N$, $N \log N$, N, N^2

Section 3. Generality, Correctness, and Searching

24. What would happen when you try to compile and run the following code?

```
public int search(Comparable[] a, Comparable target) {
    // ...
}
// ...
Object[] a = {‘‘red’’, new Book(‘‘A’’, ‘‘T’’, 123), new Integer(5)};
search(a, new Double(3.14));
```

(a) A compile-time error is generated.

(b) Compilation succeeds but a runtime error is generated.

(c) Compilation succeeds and the program runs without error.

(d) There is not enough information to decide.

25. What value should we expect the variable cmp to contain after the following code executes?

```
String s1 = ‘‘War’’;
String s2 = ‘‘Eagle’’;

int cmp = s2.compareTo(s1);
```

(a) some negative integer

(b) some positive integer

(c) -1

(d) 1

26. Which is the best choice for the comparison COMP (line 3) in the method below?

```
 1 public int search(Object[] a, Object target) {
 2    int = 0;
 3    while ((i < a.length) && (COMP)) {
 4       i++;
 5    }
 6    if (i < a.length)
 7       return i;
 8    else
 9       return −1;
10 }
```

(a) `a[i] != target`

(b) `!a[i].equals(target)`

(c) `a[i].compareTo(target) != 0`

(d) `a[i].compareTo(target) < 0`

27. What would happen when you try to compile and run the following code?

```
public int search(Comparable[] a, Comparable target) {
   // ...
}
// ...
Comparable[] a = {‘‘red’’, new Book(‘‘A’’, ‘‘T’’, 123), new Integer(5)};
search(a, new Double(3.14));
```

(a) A compile-time error is generated.

(b) Compilation succeeds but a runtime error is generated.

(c) Compilation succeeds and the program runs without error.

(d) There is not enough information to decide.

28. Given the following method and array declarations (recall that Number is the superclass of Integer), which of the following calls to the search method would generate a compile-time error?

```
public class SearchLib {
   public static <T> int search(T[] a, T target) {
      // ...
   }
   // ...
}
Integer[] ia = {2, 4, 6, 8};
Number[]  na = {2, 4, 6, 8};
```

(a) `SearchLib.<Integer>search(na, 4);`

(b) `SearchLib.<Integer>search(ia, 4);`

(c) `SearchLib.<Number>search(na, 4);`

(d) `SearchLib.<Number>search(ia, 4);`

29. Given the method signature below, what can you conclude about the parameter `a`?

```
public <T> int search(List<T> a, T target)
```

   (a) It is an instance of the `List` class.

   (b) It is declared as a `List` but is instantiated as an `ArrayList`.

   (c) It is an array whose elements are of the generic type `T`.

   (d) It is an instance of some class that implements the `List` interface.

30. What would happen when you try to compile and run the following code?

```
public int search(Object[] a, Object target) {
   // ...
}
// ...
Object[] a = {‘‘red’’, new Book(‘‘A’’, ‘‘T’’, 123), new Integer(5)};
search(a, new Double(3.14));
```

   (a) A compile-time error is generated.

   (b) Compilation succeeds but a runtime error is generated.

   (c) Compilation succeeds and the program runs without error.

   (d) There is not enough information to decide.

The next two questions refer to the following search method.

```
1 public static <T> int search(Collection<T> a, T target) {
2    Iterator<T> itr = a.iterator();
3    int i = 0;
4    while ((COMP1) && (COMP2)) {
5       i++;
6     }
7     if (i < a.size()) return i;
8     else return −1;
9 }
```

31. Which expression should go in the place of `COMP1` (line 4) above?

   (a) a.hasNext()

   (b) itr.hasNext()

   (c) !a.next().equals(target)

   (d) !itr.next().equals(target)

32. Which expression should go in the place of `COMP2` (line 4) above?

   (a) a.hasNext()

   (b) itr.hasNext()

   (c) !a.next().equals(target)

   (d) !itr.next().equals(target)

Answer Key for Version $\boxed{1}$

Section 1. Sorting

1. (d)
2. (b)
3. (a)
4. (d)
5. (b)
6. (c)
7. (c)
8. (d)
9. (a)
10. (d)
11. (a)
12. (c)
13. (b)

Section 2. Efficiency and Algorithm Analysis

14. (d)
15. (b)
16. (c)
17. (b)
18. (c)
19. (d)
20. (a)
21. (c)
22. (c)
23. (a)

Section 3. Generality, Correctness, and Searching

24. (a)
25. (a)
26. (b)
27. (b)
28. (a)
29. (d)
30. (c)
31. (b)
32. (d)