1 Analysis of Algorithms

1. Analyzing an algorithm with respect to execution time is called:
   (a) Memory usage analysis  (b) Time complexity analysis
   (c) Cost efficiency analysis  (d) Execution complexity analysis

2. Analyzing an algorithm with respect to execution time is based on:
   (a) Memory  (b) Time  (c) Money  (d) Primitive operations

3. The key issue in Big-Oh notation, the asymptotic complexity, is determined by:
   (a) The dominant term  (b) The largest constant
   (c) Solving the expression  (d) The execution time

4. Determine the Big-Oh notation of the expression $t(n) = 4n^2 + 50n + 12$:
   (a) $O(4n^2)$  (b) $O(4)$  (c) $O(n^2)$  (d) $O(n^2 + n)$  (e) $O(n)$

5. Determine the Big-Oh notation of the expression $t(n) = n^2 \log(n) + 8n^3$:
   (a) $O(n^2)$  (b) $O(n^3)$  (c) $O(n^5)$  (d) $O(n^2 \log(n))$  (e) $O(8)$

6. Indicate which of the following time complexities is the largest:
   (a) $O(n^2 \log(n))$  (b) $O(4)$  (c) $O(30n^2)$  (d) $O(n^2 + n)$  (e) $O(n)$

7. Compute the time complexity of the following algorithm:

   ```c
   1 int x, y, z = 0;
   2 for (x = 0; x < n; x++)
   3     for (y = 0; y < 3; y++)
   4         z = z + 1;
   ```
   (a) $O(3)$  (b) $O(n)$  (c) $O(n^2)$  (d) $O(3n)$  (e) $O(n \log(n))$
8. Compute the time complexity of the following algorithm:

```c
1 int x, y, z = 0;
2 for (x = 0; x < n; x+=2)
3   for (y = 0; y < n; y*=2)
4     z = z + 1;
```

(a) \(O(n)\) (b) \(O(n^2)\) (c) \(O(\log(n))\) (d) \(O(\log(n)^2)\) (e) \(O(n \log(n))\)

9. Compute the time complexity of the following algorithm:

```c
1 int x, y, z = 0;
2 for (x = 0; x < n; x++)
3   for (y = 0; y < n*n; y++)
4     z = z + 1;
```

(a) \(O(n)\) (b) \(O(n^2)\) (c) \(O(n^3)\) (d) \(O(n^4)\) (e) \(O(3^n)\)

10. Compute the time complexity of the following algorithm:

```c
1 int x, y, z = 0;
2 for (x = 0; x < n/2; x++)
3   if (x % 2 == 0)
4     z = z + 1;
```

(a) \(O(n/2)\) (b) \(O(n/4)\) (c) \(O(n)\) (d) \(O(n^2)\) (e) \(O(\log(n))\)

11. Determine the time complexity of performing `push()` on an `ArrayStack`.
(a) \(O(1)\) (b) \(O(n)\) (c) \(O(n^2)\)
(d) It depends whether or not the stack is full
(e) It depends on the new value being pushed on the stack

12. Determine the time complexity of performing `push()` on a `LinkedStack`.
(a) \(O(1)\) (b) \(O(n)\) (c) \(O(n^2)\)
(d) It depends whether or not the stack is full
(e) It depends on the new value being pushed on the stack
2 Recursion

Consider the following algorithm for the next few questions:

1 public void recMethod(int x) {
2     if (x <= 10)
3         return;
4     else {
5         System.out.println(x);
6         recMethod(x - 10);
7     }
8 }

13. Compute the number of times println() executes when we call recMethod(70);
   (a) 1  (b) 6  (c) 7  (d) 10  (e) 70

14. If we changed the operator on line 2 to be strictly less than, i.e. if (x < 10), how many
    times would println() execute when we call recMethod(70);
    (a) 1  (b) 6  (c) 7  (d) 10  (e) 70

15. Compute the number of times println() executes when we call recMethod(32);
    (a) 1  (b) 2  (c) 3  (d) 4  (e) 32

16. If we changed the operator on line 2 to be strictly less than, i.e. if (x < 10), how many
    times would println() execute when we call recMethod(32);
    (a) 1  (b) 2  (c) 3  (d) 4  (e) 32

17. Compute the number of times println() executes when we call recMethod(-70);
    (a) 0  (b) 1  (c) 6  (d) -6  (e) -70
Consider the following algorithm for the next few questions:

1 public int examineString(String s) {
2     if (s.equals(""))
3         return 0;
4     return 1 + examineString(s.substring(1));
5 }

18. What is the method `examineString()` doing?
(a) Reversing the string
(b) Checking if the string is a palindrome
(c) Counting the number of characters in the string
(d) Jumbling the characters within the string
(e) None of the above

19 What is the base case of `examineString()`?
(a) When the integer equals 0
(b) When the string equals "" (empty string)
(c) When the string has a character chopped off
(d) When the string equals "0"

20. What is the recursive case of `examineString()`?
(a) Calling `examineString()` on the string after chopping off the first character
(b) Calling `examineString()` on the string after chopping off the last character
(c) Adding 1 to a counter
(d) Adding the first character to the end of the string

21. Determine the value of \(x\) after executing the function call `int x = examineString("Apple");`
(a) "" (empty string)  (b) "Apple"   (c) 0   (d) 1   (e) 5

22. How many times is `examineString()` called when calling it on a string of \(n\) characters?
(a) 1  (b) \(n - 1\)  (c) \(n\)  (d) \(n + 1\)  (e) \(n^2\)

23. Determine the number of times `examineString()` is called in total when executing the following lines of code:

1 int x = examineString("Tom");
2 int y = examineString("Jennifer");
3 int z = examineString("Halle");

(a) 3  (b) 8  (c) 13  (d) 16  (e) 19
3 Memory Management

24. A single copy of each class and interface of the executing Java program is stored in which part of memory?
   (a) Execution stack  (b) Static heap  (c) Dynamic heap
   (d) Class memory    (e) Stack overflow

25. Class instances (objects) of the executing Java program are stored where in memory?
   (a) Execution stack  (b) Static heap  (c) Dynamic heap
   (d) Class memory    (e) Stack overflow

Consider the following code fragment within the main() method:

1 Person p;
2 p = new Person("Will", "Ferrell", "elf@northpole.ca");

26. The variable \textit{p} (declared on line 1) is stored in which part of memory?
   (a) Execution stack  (b) Static heap  (c) Dynamic heap
   (d) Class memory    (e) Stack overflow

27. The Person object (initialized on line 2) is stored in which part of memory?
   (a) Execution stack  (b) Static heap  (c) Dynamic heap
   (d) Class memory    (e) Stack overflow

28. Java will automatically flag objects that are not being referenced by any variables and allow their memory to be re-allocated. This is known as:
   (a) Memory removal  (b) Memory cleanup  (c) Garbage collection
   (d) Object de-allocation (e) Stack overflow
Consider the following algorithm for the next few questions (assume we have a `Person` class created like the one we used in class):

```java
1 public void m1(int j) {
2     System.out.println("Step 1");
3     m2(j, 4);
4     System.out.println("Step 2");
5     m3();
6     System.out.println("Step 3");
7     Person p = new Person("Johnny", "Depp", "");
8     System.out.println(p.getFirstName());
9 }
```

```java
1 public void m2(int x, int y) {
2     String name = "Tanya";
3     int z = x + y;
4     if (z > 7) name = "Zanya";
5 }
```

```java
1 public void m3(Person p) {
2     String fname = p.getFirstName();
3     String lname = p.getLastName();
4     String email = p.getEmail();
5     System.out.println(p.getLastName());
6 }
```

29. Determine the set of variables that would be included in the activation stack upon executing `m2(3, 5)`:
   (a) 3, 5, Tanya  (b) 3, 5, Zanya  (c) x, y, name  (d) x, y, z  (e) x, y, z, name

30. Determine the set of variables that would be included in the activation stack upon executing `m3(new Person("Adam", "Sandler", "adams@uwo.ca"));`
   (a) p  (b) fname, lname, email  (c) p, fname, lname, email
   (d) Adam, Sandler, adams@uwo.ca
   (e) p.getFirstName(), p.getLastName(), p.getEmail()

31. Determine the **maximum** number of activation records on the execution stack at a single moment at any time during the execution of the following code:

```java
1 public static void main(String[] args) {
2     System.out.println("Welcome!");
3     m1(15);
4     System.out.println("Have a nice day.");
5 }
```
   (a) 3  (b) 4  (c) 5  (d) 6  (e) 7 or more
4 Lists

32. Which of the following is not one of the lists discussed in class?
   (a) Ordered list   (b) Unordered list   (c) Indexed list   (d) Random list

33. Adding elements to a list works the same regardless of the list type:
   (a) True   (b) False

34. Java allows an interface to extend another interface, i.e.
   \textit{OrderedListADT extends ListADT}:
   (a) True   (b) False

35. Lists are only implemented using arrays:
   (a) True   (b) False

36. What is the time complexity of adding an element to an array-based \textit{OrderedList} as
described in class?
   (a) $O(1)$   (b) $O(\log(n))$   (c) $O(n)$   (d) $O(n^2)$

37. Which method signature does the \textit{Comparable} interface include?
   (a) \texttt{public boolean isBefore(boolean value)}
   (b) \texttt{public String isComparable(String value)}
   (c) \texttt{public int order(int value)}
   (d) \texttt{public int compareTo(T value)}
   (e) \texttt{public boolean equals(T value)}

38. How is the \textit{add()} method of the \textit{OrderedList} shown in class so short?
   (a) Its incomplete
   (b) The Java compiler fills in a lot of the blanks for us
   (c) Logic for comparing objects is located elsewhere
   (d) It doesn’t need to be long

39. What type of list is appropriate to store the rankings (first place, second place, etc.) of
   an ongoing running competition?
   (a) Ordered list   (b) Unordered list   (c) Indexed list   (e) Random list

40. Suppose we want to compare \textit{SavingsAccount} objects by their interest rate, using the
    rules discussed in class for using the \textit{Comparable} interface. If \textit{SavingsAccount} \textit{A}
    should come before \textit{SavingsAccount} \textit{B} in the proper order, which of the following
    statements would achieve that?
    (a) \texttt{if (A.getInterestRate() < B.getInterestRate()) return -1}
    (b) \texttt{if (A.getInterestRate() < B.getInterestRate()) return true}
    (c) \texttt{if (A.getInterestRate() < B.getInterestRate()) return "less"}
    (d) \texttt{if (A.getInterestRate() < B.getInterestRate()) return A.getInterestRate()}