**Data Structures and Algorithms**

**Knowledge /10**

1. Which of the following data structures **is for sure contiguous** in the memory?
2. Singly Linked List
3. ArrayList of int
4. Perfect Binary Tree
5. Priority Queue
6. Assuming you have the same data sorted in the various data structures below, which one would be the quickest in **accessing** an item at specific index?
7. Array
8. Singly Linked List
9. Doubly Linked List
10. ArrayList
11. What data structure would be best for parsing a math expression?
12. Binary Tree
13. Stack
14. Priority Queue
15. Doubly Linked List
16. Which of the following is a **typical** application of recursion?
17. Breadth-first traversal of a tree
18. Insertion Sort Algorithm
19. Finding the sum of all numbers in a DLL
20. Finding a file on the hard drive
21. Which of the structures below are trees?

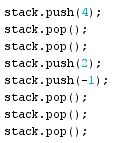
* a.)
* b.)
* c.)
* d.)
* e.)

1. Which algorithm from below performs the least number of comparisons?
2. Linear search with while-loop
3. Linear search with for-loop
4. Linear search with sentinel
5. Iterative binary search
6. Minimum time required for sorting **n** data items is:
7. **O(1)**
8. **O(n)**
9. **O(n\*log(n))**
10. **O(n2)**
11. In a Singly Linked List the head is not a node but a reference to the first node.
12. True
13. False
14. In a Doubly Linked List it is possible to have an empty node.
15. True
16. False
17. In the context of algorithm analysis, which of the following operations is **NOT** primitive?
18. **num.sort()**
19. **y = ((a + b)/c – d)%e**
20. **x == y**
21. **System.out.println(num[i]**

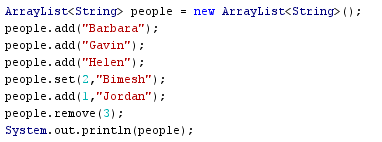
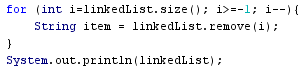
**Communication /12**

1. What is **algorithm analysis** concerned with? How are algorithms **compared**?
2. What is a **dynamic** data structure? What are the pros and cons?
3. What are the **pros and cons of a Doubly Linked List** over a Singly Linked List?
4. Give one example of a software application in which a **Priority Queue** should be used. Explain.
5. What are the **recursion pitfalls**? What measures should you take when using recursion to avoid them?
6. Briefly describe how **Mergesort** works. What type of algorithm is it – iterative or recursive?

**Thinking /5**

1. Which algorithm is **the most suitable** for sorting a large set of customer names, stored in a text file?
2. Selection sort
3. Insertion sort
4. Quicksort
5. Mergesort
6. What item is left on the top of the given **existing stack** after the following operations have been executed?

|  |
| --- |
|  |
| **3** |
| **5** |
| **1** |
| **-5** |
| **2** |

1. 3
2. 5
3. -1
4. 1
5. -5
6. 4
7. What is printed after executing the following code segment?
8. [Barbara, Jordan, Gavin]
9. [Jordan, Bimesh, Gavin]
10. [Gavin, Jordan, Bimesh]
11. ArrayIndexOutOfBoundsException is thrown
12. Draw a **perfect** Binary Search Tree that stores the following items: 9, 14, 5, 6, 19, 18, 2
13. What will be output in the console if a Java Linked List with name **linkedList** holds the String items {"RHHS","is","the","best","school"}, and the following code is executed:
14. RHHS
15. blank line
16. Null
17. error

**Thinking /10**

1. Recursive Palindrome

Design a **recursive** method called **isPalindrome()** that accepts a string as a parameter and returns true if the string is a palindrome and false otherwise. You can assume that the string is a single word (no spaces).

Examples: “noon” -> true “rotator” -> true “apple” -> false

**Note**: An empty string is considered a palindrome!

1. Add method for SLL

Write a void method called **addLast()** that adds an item to a Singly Linked List (no tail!).