# **Lab 5 Recursion**

# **Week beginning 8th October, 2018**

**Recursion**

**Use JUnit for all the methods which have a return value. Otherwise test by hand in main() method.**

1. Step through the code for Factorial method in debug mode and see all the method calls and the methods returning.

2. Write a recursive version of pow function i.e.

public static int myPow(int x, int y) where y an integer >= 0 // Returns the value of the first argument raised to the power of the second argument.

Use Debug mode to see the recursive steps when the method is called.

3. Write an iterative version of myPow method. Profile both iterative and recursive version. Which is more efficient in terms of CPU time?

4. Write a recursive method to find the smallest element in an array.

Hint: find the smallest element in the partial array containing all but the last element. Then compare that smallest to the value of the last element.

5. Write a recursive version of binary search method. See code for binarySearch method in BinarySearch folder

6. CP3LinkedList has a print method that outputs the data on the list. Write a recursive version of this method. It will call the following helper method:

private void printSub(Node head)

Hint: If head is null, then there’s nothing to output. Otherwise, output the head.data before recursively moving on to head.next.

Difficult: Then consider how you could change the printSub method to output the nodes in reverse order.

7. The Fibonacci sequence used in Mathematics is defined as follows:

F(0) = 0

F(1) = 1

F(n) = F(n-1) + F(n-2) where n > 1

i.e. a recursive definition.

Write a recursive and a non-recursive method to compute this function.

Debug the recursive to see the different method calls.

Can you explain why recursion is very inefficient here?

For all the methods above, consider whether an iterative or recursive method is more efficient.