# Homework 1: Analysis and design of algorithms

1. (a) What is meant by the **time complexity** of an algorithm? [2]

(b) The time complexity of an algorithm can be expressed in Big-O notation. Why is this measure of interest to computer scientists? [2]

(c) Time complexity can be roughly calculated by counting the number of steps involved in processing n items.

If the number of steps in an algorithm is computed as 3n3 + n2 + 10n + 6, what is the Big-0 notation that expresses the time complexity of the algorithm? Explain how you arrive at your answer. [3]

2. (a) Arrange the following in ascending order of time complexity:

O(n2) O(log n) O(n!) O(n) O(2n) [3]

(b) For each one, state whether it is constant time, exponential time, logarithmic time, polynomial time or linear time. [5]

(i) O(n2)

(ii) O(log n)

(iii) O(n!)

(iv) O(n)

(v) O(2n)

3. The algorithm for a bubble sort of n items is given below.

for i = 1 to n-1

for j = 1 to (n-i)

if numbers [j] > numbers[j+1]

# Swap the names in the array

temp = numbers[j]

numbers[count] = numbers[count+1]

numbers[count+1] = temp

endif

next j

next i

(a) Calculate the number of steps needed to sort a list of 5 items.   
Count the IF statement in the inner loop as one statement. [2]

(b) Simplify the number of steps to sort a list of n items, given that

1 + 2 +… + n-1 = n(n-1)/2 [2]

(c) What is the time complexity of the algorithm expressed in Big-O notation? [1]

Total 20 marks