Module IN2002—Data Structures and Algorithms
Exercise Sheet 1

Note that you may not have enough time to finish this during the tutorial slot. Get started, see where you have difficulties, study, ask questions, and try again. You should aim to finish all the questions before the next lecture.

1. Consider the following pseudocode :

```
Function foo(array)
i ← 1
WHILE array[i] mod 23 != 0 AND i <= length of array
        i ← i+1
Return i
```

(a) Describe in plain English what this algorithm does.
(b) What is the time complexity with respect to the array size n of the algorithm described in the worst case, best case, average case? Justify your answer and state assumptions you make for the average case.

2. Suppose a program consists of two phases executed one after the other. The program takes a string as input, which may be of any length $n \in \mathbb{N}$. What is the time complexity of the whole program, with the phases being of the following complexities?
(a) $O(n^2)$ and $O(n \log n)$
(b) $O(\log n)$ and $O(n)$
(c) $O(n)$ and $O(n)$

3. Consider the following pseudocode:

```
Function foo(k)
s ← 1
WHILE k > 0
        s ← s * 2
        k ← k-1
Return s
```

(a) What does *foo* compute?
i. nothing
ii. k^2
iii. s * k
iv. 2^k

(b) What is the time complexity of *foo* with respect to *k* in the worst case?
i. O(1)
ii. $O(2^k)$
iii.O(s)
iv. O(k)

(c) How is the complexity different for the best or average case?
Justify your answers.

4. Create an algorithm that given an array of integers returns the number of times that the number 100 is present in the array. Then answer the questions below.

(a) What is the time complexity of your algorithm? Justify your answer.
(b) What is its space complexity? Justify your answer.
(c) Would it make a difference if the input array was sorted? Justify your answer.

5. Consider the following pseudocode:

```
Function bar(k)
WHILE k > 0
        i ← k
        s ← 0
        WHILE i > 0
                s ← s + i
                i ← i -1
        array[k] ← s
        k ← k – 1
Return array
```

(a) What does *bar* compute?
i. []
ii. [1,2,3, ... , k]
iii. [1,1+2,1+2+3, ... , 1+ ... +k]
iv. k * s

(b) What is the time complexity of *bar* with respect to *k* in the worst case:
i. O(1)
ii. O(s)
iii. O(k)
iv. $O(k^2)$

(c) Is the complexity different for the best or average case?

(d) How could the computation in *bar* be done in a more efficient way? (You need to remember your Maths knowledge to do this. This link will help: http://www.cut-the-knot.org/Curriculum/Algebra/GaussSummation.shtml). Justify your answers.