THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

COMPSCI.220.S1 – Algorithms and Data Structures

Assignment 1 – Math Basics & Time Complexity of Algorithms

Out: Friday, 2nd of March, 2018

Due: Friday, 16th of March, 2018

Assignment 1 (Lectures 1 – 7) is worth **75 marks** representing **7.5%** of your total course grade.

Objectives. Learning basic math techniques (sets, logarithms, exponential functions, inequalities, sequences, etc) and ways to analyse time complexity, in particular, roughly evaluate running time of a certain pseudocode and explore performance of an algorithm with "Big-Oh", "Big-Omega", and "Big-Theta" tools.

Requirements. You should answer in detail to the following questions:

1. (*8 marks*) Write down a set $S_{\mathrm{g}} = \{g_1, g_2, \ldots\}$ of letters, g_i; $i = 1, 2, \ldots$, appearing in your own given name, and a set $S_{\mathrm{f}} = \{f_1, f_2, \ldots\}$ of letters, f_i; $i = 1, 2, \ldots$, appearing in your own family name.

   Note: The name should be the same as in your UoA ID card, but with only lower-case English letters, like, e.g., "miika" instead of "Miika". If you have two or more given names, like, e.g., "anna sophia", or your family name consists of two or more words, e.g., "petrov-vodkin" or "middleton windsor", take only the first one, i.e., "anna", "petrov", or "middleton" in the above examples.

2. (*8 marks*) Find the union, $S_{\mathrm{u}} = S_{\mathrm{g}} \bigcup S_{\mathrm{f}}$, and the complement, $S_{\mathrm{c}} = S_{\mathrm{g}} \setminus S_{\mathrm{f}}$, of the sets formed in Question 1.

3. (*8 marks*) Work out the exact time complexity $T(n)$ (e.g., $T(n) = 5Cn^2 + 1$) of the following piece of code in terms of the number of operations:

```
for ( int i = 0; i < n; i += 5 ) {
     for ( int j = 1; j < n; j *= 3 ) {
          for ( int k = 1; k < n; k *= n ) {
               // constant number C of operations
          }
     }
}
```

4. (*8 marks*) Work out the exact time complexity $T(n)$ of the following piece of code in terms of the number of operations:

```
for ( int k = n; k > 1; k /= 3 ) {
     for ( int i = 0; i < n; i += 2 ) {
          // constant number C of operations
     }
     for ( int j = 2; j < n; j = ( j * j ) ) {
          // constant number C of operations
     }
}
```

5. (*8 marks*) You have found empirically that the implemented sorting methods A of complexity $\Theta(n^3)$ and B of complexity $\Theta(n^2 \log_{10} n)$ spent 2 and 10 time units, respectively, to sort an array of 100 objects. Find out how many time units will each algorithm spend for sorting an array of $1,000,000$ objects?

6. (*9 marks*) Prove that $T(n) = 5n \log_2 n + 500n$ is $\Omega(n)$ and $O(n^{1+\varepsilon})$ where $\varepsilon > 0$ is an arbitrary small positive constant.

7. (*8 marks*) You have theoretically derived that the processing time $T(n)$ of a certain algorithm is both $\Omega(n)$ and $O(n^3)$. Decide whether you can conclude that $T(n)$ is $\Theta(n^2)$. Explain your decision.

8. (*9 marks*) Let the processing time $T(n)$ of a certain optimisation algorithm depend on the problem size n as follows: $T(n) = 0.01n^2 \log_2 n + 6n(\log_2 n)^2$. Determine the "Big-Theta" time complexity bounds for this algorithm, decide whether the statements "$T(n)$ is $O(n^3)$" and "$T(n)$ is $\Omega(n^2)$ hold for this algorithm, and prove your decisions.

9. (*9 marks*) Assuming $n = 5^m$ with the integer $m = \log_5 n$, derive a closed-form formula for $T(n)$ by solving the recurrence $T(n) = 5T(n/5) + 5$ with the base condition $T(1) = 0$.

Submission: Submit your report with detailed answers to Questions 1–9 as a single Adobe PDF file to Canvas. ***Scanned handwritten documents are strictly forbidden*** (even as images in a pdf file) and will not be accepted for marking. Note that the due date is Friday, 16$^{\text{th}}$ of March 2018, 23:59. If submitted after the due date, the penalty of 10% will be before 17$^{\text{th}}$ of March 2018, 23:59; then the penalty of 50% will be before 18$^{\text{th}}$ of March 2018, 23:59, and no submission afterwards.