Special instructions: You may work either individually or in a group of exactly two persons. Write your solutions out on paper, photocopy them and hand in both the original and the copy to SSO by 12:00 on Friday, 21st October, 2016. If you are working in a group, one script should be submitted for the group, with both names at the top; both members of the group will then receive the same mark. Clearly write your name(s), student ID number(s) and the words "Comp36111 Sec. A Coursework" on the front (cover) sheet and staple all sheets together. Submitted solutions must be entirely the effort of the persons whose names appear at the top of the submission.

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Advanced Algorithms I: Coursework for Sec. A

Time: This should take you a few hours

Please answer all questions.

Marks will be awarded for clarity and succinctness as well as correctness.

The use of electronic calculators is not recommended.

Section A

1. As usual, we denote the natural numbers $\{0, 1, \ldots\}$ by $\mathbb{N}$. In Lecture1, we encountered the LOOP programming language, whose constructs are

   `x= y,   x= 0,   x++,   return x,   loop(x) {` $\cdots$ `}`

   with the expected semantics, assuming that the variables `x`, `y`, ..., take values in $\mathbb{N}$. (Note that `loop(x){ ...}` loops x number of times, where x is the contents of `x` on initial entry to the loop, even if the contents of `x` is changed in the loop body.)

   Write LOOP programs to compute the functions $x \mapsto x \dot{-} 1$ and $x, y \mapsto x \dot{-} y$, where $x \dot{-} y$ is defined to be $x - y$ if $y \leq x$, and 0 otherwise. Assume that the input values x and y are stored in the respective registers `x` and `y` at the start of execution. Endeavour to minimize the depth of nesting of `loop` constructors. (4 marks)

2. Let $g : \mathbb{N} \to \mathbb{N}$ be a function. Explain informally why the running time of any LOOP program computing the function g on input x must be at least $g(x) \dot{-} x$. (2 marks)

3. If $f : \mathbb{N} \to \mathbb{N}$ is a function, we take $f^{(k)}(x)$ to denote the k-fold iteration $f(f(\cdots f(x)))$ of f applied to x. In the special case $k = 0$, we take $f^{(0)}(x)$ to denote x.

   Now define the sequence of functions $f_n : \mathbb{N} \to \mathbb{N}$, for $n \in \mathbb{N}$, as follows:

   $$f_0(x) = \begin{cases} x+1 & \text{if } x \leq 1 \\ x+2 & \text{otherwise} \end{cases} \qquad\qquad f_{n+1}(x) = f_n^{(x)}(1).$$

   Write a simpler definition of the function f_1. (2 marks)

4. Give an inductive proof that, for all n, f_n is *increasing*: $x < y \Rightarrow f_n(x) < f_n(y)$. Prove in addition that, for all n, f_n is *expansive*: $x < f_n(x)$. (4 marks)

5. Show that, for all $n \geq m$, and all x, $f_n(x) \geq f_m(x)$. Deduce that, for all $k \in \mathbb{N}$ and $n \geq 1$, $f_n^{(k+1)}(x) > f_n^{(k)}(x) + x$. (2 marks)

6. Denote by $\mathcal{L}_n$ the class of functions $f : \mathbb{N} \to \mathbb{N}$ definable by LOOP programs with loops nested to depth at most n. Prove that, for all $n \geq 1$, $f_n \in \mathcal{L}_n$. (4 marks)

7. With relatively little work, it can be shown that, for all n, *any* program P with loops nested at most to level at most n has running time bounded by $f_n^{(k)}(x)$, where x is the maximum value of its inputs. (Here k depends on P alone.) Further, it can be shown that, for all n and k, there exists x_0 such that $x \geq x_0$ implies $f_{n+1}(x) \geq f_n^{(k)}(x)$. Assuming these results, show that $\mathcal{L}_n$ is a *proper* subset of $\mathcal{L}_{n+1}$. (2 marks)

END OF EXAMINATION