Note: Please deposit your solutions in the appropriate box **by 4 p.m. on the due date**. Late assignments or assignments placed into incorrect boxes will not be marked. Use a cover sheet available from outside the Resource Centre.

PLEASE SHOW ALL WORKING.

1. The following algorithm for evaluating a polynomial is called **Horner's rule**. The polynomial $a_0 + a_1 x + \cdots + a_n x^n$ is represented by the array $A[0..n]$, where $A[i] = a_i$ for each i.

algorithm `horner`(integer array $A[0..n]$)
$val := A[n]$
for j from $n-1$ downto 0 **do**
    $val := val * x + A[j]$
return val;
end

Prove that Horner's rule is correct.

2. Describe the worst-case running time of `russmult` in the case when $x \leq y$ and in the case when $y \leq x$. Your answer should involve the Θ notation. Give all details. Note: remember that addition, doubling and halving cannot all be considered as constant time operations!

3. Arrange the following functions in ascending order of growth rate. That is, if g follows f in your list, then we must have $f \in O(g)$. Give full explanation.

- $f_1(n) = n^{5/2}$
- $f_2(n) = 2^{2^n}$
- $f_3(n) = \sqrt{2n}$
- $f_4(n) = n + 10$
- $f_5(n) = 2^{\sqrt{\log n}}$
- $f_6(n) = n^{\log n}$
- $f_7(n) = 2^{n^2}$
- $f_8(n) = n^n$
- $f_9(n) = n!$
- $f_{10}(n) = n(\log n)^3$

4. Show by explicit counterexamples that if any of the 3 hypotheses in the smoothness rule is removed, then the result is no longer true. That is, for each of the conditions: f is eventually increasing; g is eventually increasing; g has subexponential growth, you should give an example of f and g where that condition does not hold but the other two conditions do, and $f(n) \in O(g(n))$ when n is a power of 2, but $f(n) \notin O(g(n))$.

5. An **Egyptian fraction** is a rational number of the form $1/n$ where n is a positive integer. Give a greedy algorithm that given as input a positive rational number p/q, will find a representation of r as a sum of Egyptian fractions, and prove its correctness. Show how your algorithm would work on the input $11/37$.

Note: your algorithm will probably not be optimal in the sense of finding the smallest number of summands, but that is OK.

6. Suppose that you have to post a letter and have an unlimited supply of stamps worth $1, 5, 10, 20, 50$ units. Give a greedy algorithm that solves this problem for an arbitrary positive integer postage value n. Prove that it is correct. Prove that it is also optimal if you want to minimize the number of stamps used.

7. Suppose that instead you have stamps worth $1, 3, 6, 12, 24, 30$ units. Is your algorithm above still optimal in the sense of using the fewest stamps?

8. Consider the following algorithm to find a minimum weight spanning tree of a connected graph. Start with the full graph and repeatedly delete edges in decreasing order of cost. The only time an edge is not deleted is if doing so would disconnect the graph. Stop when there are no more edges that can be deleted safely. Is this algorithm correct? If so, give a proof, and if not, give a counterexample.