The answers must be word-processed or typed. You may substitute formulae and figures with hand-writings. Your submitted algorithms should be in the pseudo-code, not in any specific programming language. Answers deviating from these requirements will be returned without grading.

The answers must be the student’s own work. Idea sharing and referencing to others’ work (including those online) are not allowed. Plagiarism and other forms of academic dishonesty will be handled within the guidelines of the Student Handbook and reported to the University.

There are 7 questions and 120 points in total.

1. (15 points) Consider the following iterative algorithm in pseudo-code:

```plaintext
Algorithm DoSomething (A[1..n]: a list of integers);
    n = length(A);
    repeat
        flag = FALSE;
        for i = 2 to n do
            if A[i-1] > A[i] then
                x = A[i-1];
                A[i-1] = A[i];
                A[i] = x;
                flag = TRUE;
        until not flag;
    return (A);
```

Rewrite algorithm `DoSomething` where both the loop statements `repeat...until` and `for` are replaced with recursion statements. Make sure your
rewritten algorithm accomplishes exactly the same task as DoSomething does.

2. (15 points) Consider the problem that a turtle crawls in straight line toward a pond one mile away. On day one, it crawls one-half mile, on the following days, it crawls one half of the distance it has crawled the day before. This question asks you to give a recurrence formula to account for the remaining distance the turtle is from the pond on the \(k\)th day, for any \(k \geq 1\). Note that you may design and utilize more than one recurrence formula.

3. (15 points) Question 3-2, page 61 (3rd ed), page 58 (2nd ed). Complete only columns of \(O\) and \(\Omega\) in the table.

4. (15 points) Questions 3-4: page 62 (3rd ed), page 59 (2nd ed). Answer questions for \(a, d, e\) only.

5. (20 points) Use a recursion tree to determine a good asymptotic upper bound on the recurrence \(T(n) = T(n/2) + n^2\). You may assume that \(n\) is always a power of 2 and a base case \(T(1) = c\) for some constant \(c > 0\).

6. (20 points) Use the substitution method to verify your answer for the above Q5.

7. (20 points) Derive a good upper bound for recurrence \(T(n) = 5T(n/3) + n\), with base case \(T(n) = c\) when \(n \leq 3\), for some constant \(c > 0\).

The answers must be word-processed or typed. You may substitute formulae and figures with hand-writings. Your submitted algorithms should be in the pseudo-code, not in any specific programming language. Answers deviating from these requirements will be returned without grading.

The answers must be the student’s own work. Idea sharing and referencing to others’ work (including those online) are not allowed. Plagiarism and other forms of academic dishonesty will be handled within the guidelines of the Student Handbook and reported to the University.