Assignment 1

Grades: each assignment is 5% in the final score and there is a 2% bonus in each assignment for bonus problems. We use the following scale for the simplicity of grading: the total score is 50 (additional 20 for bonus problems) for each assignment.

Problem 1 [Full Score: 15]. Solve the following problems from the book. For each problem, provide the "big-Oh" characterization of the running time and a brief explanation. You do not need a formal proof.

1) [5] R-1.11 (on page 48).
2) [5] R-1.13
3) [5] R-1.14

Problem 2 [Full Score: 15]. Prove the following properties about the "big-Oh" notation from the textbook. Assume \( f(n), g(n), d(n), e(n), h(n) \) are non-negative. Remember to show your choices of \( c \) and \( n_0 \) for each \( O(\cdot) \) statement.

2) [5] R-1.16
3) [5] R-1.25

Problem 3 [Full Score: 20]. As all for loop algorithms, Algorithm 1.2 (p.7) can be written as a while loop. Write such a loop \( L \): while \( C \) do \( B \) for Algorithm 1.2. Then design a loop invariant \( I \) to prove the correctness of the computation which converts the before-loop state \( \langle P \rangle \) into the after-loop state \( \langle Q \rangle \): \( P \) is \((i = 0) \land (currentMax = A[0])\) and \( Q \) is "currentMax is the maximum value stored in \( A \)". Prove the correctness of the while loop by proving the following three conditions: (follow the examples in the note)

- \( P \rightarrow I \).
- \( (I \land C) \land B \langle I \rangle \).
- \( (I \land \neg C) \rightarrow Q \).

Problem 4 [Bonus Problems: 20]. Solve the following problems from the textbooks about \( O(\cdot) \) and \( \Omega(\cdot) \) notations.

1) [10] C-1.9 (on page 51)
2) [10] C-1.11