Discussion 2

1. Consider the following pseudocode.

   procedure AlgorithmX(n > 1)
   1. for i := 1 to n
   2.   Statement A.
   3. for j := 1 to n
   4.   Statement B.

   procedure AlgorithmY(n > 1)
   1. for i := 1 to n
   2.   for j := 1 to n
   3.   Statement C.

   How many times are statements A, B and C executed? What is the order of AlgorithmX and AlgorithmY in O notation?

2. Suppose that an element is known to be among the first four elements in a list of 32 elements. Would a linear search or a binary search locate this element more rapidly? (cf. Rosen 3.3 Exercise 7)

3. Consider the following pseudocode.

   \[
   \begin{align*}
   i &:= 1 \quad (1) \\
   t &:= 0 \quad (2) \\
   \text{while } i \leq n \quad (3) \\
   &\quad t := t + i \quad (4) \\
   &\quad i := 2i \quad (5)
   \end{align*}
   \]

   Count the number of operations (as a function of n) of this program, where an operation is an addition or a multiplication (ignore the comparisons used to test the conditions in the while loop).

   (cf. Rosen 3.3 Exercise 4)

4. Arrange the functions \((1.5)^n, n^{100}, (\log n)^3, \sqrt{n} \log n, 10^n, (n!)^2, n^{99} + n^{98}\) in a list so that each function is big-O of the next function.

   (cf. Rosen 3.2 Exercise 22)
5. Consider the following pseudocode.

**procedure** Statements($n > 1$)

1. for $i := 1$ to $10$
2. 
   *Statement A.*
3. for $j := 1$ to $n$
4. 
   *Statement B.*
5. for $k := 1$ to $4$
6. for $\ell := 1$ to $n$
7. 
   *Statement C.*

Which statement (A, B, or C) is executed the most number of times?

Suppose that Statement A requires $3n$ comparison operations, Statement B requires $n^2$ comparisons, and Statement C requires 30 comparisons. How many total comparisons does the entire pseudocode segment require? What is the order of this algorithm in $\Theta$ notation?