Discussion 2

1. 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)

2. Consider the following pseudocode.

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

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 \textbf{while} loop).

(cf. Rosen 3.3 Exercise 4)
3. 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)

4. Consider the following pseudocode.

procedure Statements\(\(n > 1\)\)
1. for \(i := 1\) to 10
2. \hspace{0.5cm} Statement A.
3. for \(j := 1\) to \(n\)
4. \hspace{0.5cm} Statement B.
5. for \(k := 1\) to 4
6. \hspace{0.5cm} for \(\ell := 1\) to \(n\)
7. \hspace{1.5cm} 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?