Reading (from the textbook): Sections 2.1, 2.2, 2.4, 5.1, 5.2, 5.5

Problems
(On this homework only, the problems from the book are written out.)

1.) pg. 67, problem 3 (Show all work)
Take the following list of functions and arrange them in ascending order of growth rate. That is, if function \( g(n) \) immediately follows function \( f(n) \) on your list, then it should be the case that \( f(n) \) is \( O(g(n)) \).

\[
\begin{align*}
  f_1(n) &= n^{2.5} \\
  f_2(n) &= \sqrt{2n} \\
  f_3(n) &= n + 10 \\
  f_4(n) &= 10^n \\
  f_5(n) &= 100^n \\
  f_6(n) &= n^2 \log n
\end{align*}
\]

2.) pg. 67, problem 4 (Show all work)
Take the following list of functions and arrange them in ascending order of growth rate. That is, if function \( g(n) \) immediately follows function \( f(n) \) on your list, then it should be the case that \( f(n) \) is \( O(g(n)) \).

\[
\begin{align*}
  g_1(n) &= 2^{\log n} \\
  g_2(n) &= 2^n \\
  g_3(n) &= n^{4/3} \\
  g_4(n) &= n(\log n)^3 \\
  g_5(n) &= n^{\log n} \\
  g_6(n) &= 2^{2^n} \\
  g_7(n) &= 2^{n^2}
\end{align*}
\]
3.) Suppose you are choosing between the following three algorithms for your problem:

- Algorithm A solves problems by dividing them into five subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time (linear time is $O(n)$)
- Algorithm B solves problems of size $n$ by recursively solving two subproblems of size $n - 1$ and then combining the solutions in constant time (constant time is $O(1)$)
- Algorithm C solves problems of size $n$ by dividing them into nine subproblems of size $n/3$, recursively solving each subproblem, and then combining the solutions in $O(n^2)$ time.

What are the running times of each of these algorithms (in big-O notation), and which would you choose?

4.) How many lines, as a function of $n$ (in $\Theta(n)$ form), does the following program print? Write a recurrence and solve it. You may assume $n$ is a power of two.

```plaintext
f(n)
  if n > 1 do
    print("still going")
    f(n/2)
    f(n/2)
```