Assignment 1
50 points

Problem 1. Suppose a polynomial in n of degree d has the form

$$p(n) = \sum_{i=0}^{d} a_i n^i$$

with leading coefficient $a_d > 0$, and let $k \geq 1$ be a constant (not necessarily an integer).

Prove the following:

(a) If $k \geq d$ then $p(n) \leq O(n^k)$.
(b) If $k \leq d$ then $p(n) \geq \Omega(n^k)$.
(c) If $k = d$ then $p(n) = \Theta(n^k)$.

Hint. To compare $p(n)$ with n^k, it is insightful to look at the quotient:

$$\frac{p(n)}{n^k} = \sum_{i=0}^{d} a_i n^{i-k}$$

and divide into cases from there.

If $k > d$, all of the terms on the right have negative exponents, so the right side tends to 0 as $n \to \infty$. In particular, there is an integer n_0 such that the right side is less than 1 for $n > n_0$, and therefore

$$p(n) <= n^k \qquad \text{for } n > n_0.$$

This leads directly to (a).

If $k = d$, we can group the terms as follows:

$$\frac{p(n)}{n^k} = \sum_{i=0}^{d-1} a_i n^{i-d} + a_n$$

and note that the sum tends to 0 as $n \to \infty$, so that there is an n_1 such that

$$a_n - 1 \leq \frac{p(n)}{n^d} <= a_n + 1 \qquad \text{for } n > n_1$$

or

$$(a_n - 1)n^d \leq p(n) <= (a_n + 1)n^d \qquad \text{for } n > n_1.$$

This leads directly to (b). And so on.

Problem 2. Prove that $\log n! = \Theta(n \log n)$ without using Stirling's formula.

Hint. Prove these two factoids using the definition of factorial:

$$n! < n^n$$
$$n! > \left(\frac{n}{2}\right)^{\frac{n}{2}}$$

That is, the two expressions on the right are upper and lower bounds of $n!$, respectively. Then show that each of these bounds is $\Theta(n \log n)$.

Problem 3. Show that the generic algorithm `g_set_intersection` has runtime $O(n)$, where n is the number of items in the two ranges being intersected. Begin by stating the algorithm, including its assumptions, asserted outcomes, and body. After proving the runtime is $O(n)$, explain why it is *not* $\Omega(n)$ (or $\Theta(n)$).

Problem 4. Devise three programming scenarios, one for each of the three sequential container classes Vector, List, and Deque, in which the choice of that container is the best (or only) choice among the three. Give complete arguments for the choices in each scenario.

Problem 5. Among the key-comparison sort algorithms, explain why Merge Sort is the best choice to sort a linked list, but not for an array. Then give the pros and cons for the other key-comparison sorts when applied to an array (or vector or deque).