Exercise 1

Let log denote the natural logarithm (i.e., base e). Order the following functions by their order of growth, from slowest to fastest:

1. 2^{2^n}
2. $e^{\log n}$
3. $\log(n)$
4. e^n
5. $e^{2\log(n)}$
6. $\log(\log(n))$
7. 2^n
8. $n!$

Solution: Ordered by Growth Rate (Slow to Fast)

- $\log(\log n)$: double logarithmic, grows the slowest
- $\log n$: logarithmic
- $e^{\log n} = n$: because $e^{\log n} = n$
- $e^{2\log n} = n^2$: using $e^{2\log n} = (e^{\log n})^2 = n^2$
- 2^n: exponential base 2
- e^n: exponential base e, grows faster than 2^n
- $n!$: super-exponential (Stirling's approx: $n! \approx n^n$)
- 2^{2^n}: double exponential, fastest

$$\log(\log n) \prec \log n \prec n \prec n^2 \prec 2^n \prec e^n \prec n! \prec 2^{2^n}$$

Exercise 2: Prove Big-O Properties

Let $f, g, h : \mathbb{N} \to \mathbb{R}_{\geq 0}$. Prove the following:

1. $f \in \mathcal{O}(f)$

   Proof: By definition, $f(n) \leq c \cdot f(n)$ for $c = 1$ and all $n \geq 1$. Hence, $f \in \mathcal{O}(f)$.

2. If $c > 0$, then $\mathcal{O}(c \cdot f) = \mathcal{O}(f)$

   Proof: If $f(n) \leq c_1 \cdot g(n)$, then $cf(n) \leq (cc_1) \cdot g(n)$. Constants scale but do not affect asymptotic class.

3. If $f(n) \leq g(n)$ for sufficiently large n, then $\mathcal{O}(f) \subseteq \mathcal{O}(g)$

   Proof: Given $f(n) \leq g(n)$ for all $n \geq n_0$, then $f(n) \leq 1 \cdot g(n) \Rightarrow f \in \mathcal{O}(g)$. So any function in $\mathcal{O}(f)$ grows no faster than g, and thus is in $\mathcal{O}(g)$.

4. If $\mathcal{O}(f) \subseteq \mathcal{O}(g)$, then $\mathcal{O}(f + h) \subseteq \mathcal{O}(g + h)$

   Proof: Let $f(n) \leq c \cdot g(n)$. Then:

   $$f(n) + h(n) \leq c \cdot g(n) + h(n) \leq \max(c, 1) \cdot (g(n) + h(n))$$

   So $f + h \in \mathcal{O}(g + h)$.

5. If $h(n) > 0$ and $\mathcal{O}(f) \subseteq \mathcal{O}(g)$, then $\mathcal{O}(f \cdot h) \subseteq \mathcal{O}(g \cdot h)$

   Proof: From $f(n) \leq c \cdot g(n)$, multiply both sides by $h(n) > 0$:

   $$f(n) \cdot h(n) \leq c \cdot g(n) \cdot h(n)$$

   Therefore, $f \cdot h \in \mathcal{O}(g \cdot h)$.

Exercise 3: Growth Comparison of Polynomials

Let $i, j, k, n \in \mathbb{N}$. Prove:

1. If $j \leq k$, then $\mathcal{O}(n^j) \subseteq \mathcal{O}(n^k)$

   Proof: Since $n^j \leq n^k$ for all $n \geq 1$, the result follows directly.

2. If $j \leq k$, then $\mathcal{O}(n^j + n^k) \subseteq \mathcal{O}(n^k)$

   Proof: $n^j + n^k \leq 2n^k$ for large n, so this is in $\mathcal{O}(n^k)$.

3. $\mathcal{O}\left(\sum_{i=0}^{k} a_i n^i\right) = \mathcal{O}(n^k)$

   Proof: The highest-degree term dominates. Lower-order terms vanish asymptotically.

4. $\mathcal{O}(\log n) \subseteq \mathcal{O}(n)$

   Proof: $\log n \leq n$ for all $n \geq 1$.

5. $\mathcal{O}(n \log n) \subseteq \mathcal{O}(n^2)$

   Proof: For all $n \geq 2$, $\log n \leq n$, so $n \log n \leq n^2$.

Exercise 4: Compare Growth Classes

1. $\mathcal{O}(n) \supset \mathcal{O}(\sqrt{n})$

   Proof: $\sqrt{n} < n$ for $n > 1$

2. $\mathcal{O}(n^2) \subset \mathcal{O}(2^n)$

   Proof: Exponential grows faster than any polynomial.

3. $\mathcal{O}(\log n) \subset \mathcal{O}(\log^2 n)$

   Proof: $\log^2 n = \log n \cdot \log n > \log n$

4. $\mathcal{O}(2^n) \subset \mathcal{O}(3^n)$

   Proof: $2^n < 3^n$

5. $\mathcal{O}(\log_2 n) = \mathcal{O}(\log_3 n)$

   Proof: Change-of-base formula shows $\log_b n = \frac{\log n}{\log b}$. All differ by a constant factor.

Exercise 5: Sorting Algorithm Runtime Comparison

- **Bubble Sort vs Insertion Sort:** Both are $\mathcal{O}(n^2)$, but insertion sort is typically faster in practice.
- **Insertion Sort vs Merge Sort:** Merge sort is $\mathcal{O}(n \log n)$, faster than insertion sort's $\mathcal{O}(n^2)$
- **Merge Sort vs Quick Sort:** Both are $\mathcal{O}(n \log n)$ on average, but quick sort has worse worst-case $\mathcal{O}(n^2)$ unless optimized.