Asymptotic Analysis

- Refers to the study of an algorithm as the input size “gets big” or reaches a limit (in the calculus sense)

- Growth rate
  - rate at which the cost of an algorithm grows as the size of its input grows
  
  \[
  \text{growth of } T(n) \text{ as } n \to \infty
  \]

- Example: \(c_1 n \text{ vs. } c_2 n^2\)

Variations in HW/SW do not alter growth rate of \(T(n)\) — only by a constant factor.

Growth Rate

\[
\begin{align*}
T(n) &= \Theta(2^n) \\
T(n) &= \Theta(n^3) \\
T(n) &= \Theta(n^2) \\
T(n) &= \Theta(n \log n) \\
T(n) &= \Theta(n) \\
T(n) &= \Theta(\log n) \\
T(n) &= \Theta(1)
\end{align*}
\]
Growth Rate

$T(n) = \Theta(2^n)$
$T(n) = \Theta(n^3)$
$T(n) = \Theta(n^2)$
$T(n) = \Theta(n \log n)$
$T(n) = \Theta(n)$
$T(n) = \Theta(\log n)$
$T(n) = \Theta(1)$

Asymptotic Performance

- For large values of $n$, a $\Theta(n^2)$ algorithm always beats a $\Theta(n^3)$ algorithm.

In practice, we shouldn’t completely ignore asymptotically slower algorithms.

Big O

Definition. $T(n) = O(f(n))$ if there are positive constants $c$ and $n_0$ such that $0 \leq T(n) \leq c \cdot f(n)$ when $n \geq n_0$

Prove that ...

- $7n - 2$ is $O(n)$
- $20n^3 + 10n \log n + 5$ is $O(n^3)$
- $3 \log n + \log \log n$ is $O(\log n)$
- $2^{100}$ is $O(1)$
**Definition.** \( T(n) = \Omega(f(n)) \) if there are positive constants \( c \) and \( n_0 \) such that \( T(n) \geq c \cdot f(n) \geq 0 \) when \( n \geq n_0 \)

**Definition.** \( T(n) = \Theta(f(n)) \) if there are positive constants \( c_1, c_2 \) and \( n_0 \) such that \( 0 \leq c_1 \cdot f(n) \leq T(n) \leq c_2 \cdot f(n) \) when \( n \geq n_0 \)

**Prove that ...**

\[
3 \log n + \log \log n \text{ is } \Omega(\log n)
\]

\[
3 \log n + \log \log n \text{ is } \Theta(\log n)
\]

**In practice ...**

“ignore constants and drop lower order terms”
## True or False?

\( \{n^2, n^4, \log n, \ldots\} \)

<table>
<thead>
<tr>
<th>Expression</th>
<th>Big O</th>
<th>Big Omega</th>
<th>Big Theta</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^2 + 3000n + 10 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 21 \log n )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 500 \log n + n^4 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \sqrt{n} + \log n^{50} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 4^n + n^{5000} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 3000n^3 + n^{3.5} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^5 + n! )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>