1 Asymptotic notations

In the analysis of algorithms, we are usually interested in how the performance of our algorithm changes as the problem size increases.

The primary tools for measuring the growth rate of a function that describes the run time of an algorithm are the asymptotic notations.

This provides a way of studying the algorithms themselves, independent of any specific hardware, operating system, compiler, programmer, etc.

2 Big-\(O\)

\[ O(g(n)) = \{ f(n) \mid \text{there exist positive constants } c \text{ and } n_0, \]
\[ \text{such that } 0 \leq f(n) \leq cg(n) \text{ for all } n \geq n_0. \} \]

3 Big-\(\Omega\)

\[ \Omega(g(n)) = \{ f(n) \mid \text{there exist positive constants } c \text{ and } n_0, \]
\[ \text{such that } 0 \leq cg(n) \leq f(n) \text{ for all } n \geq n_0. \} \]

4 Big-\(\Theta\)

\[ \Theta(g(n)) = \{ f(n) \mid \text{there exist positive constants } c_1, c_2, \text{ and } n_0, \]
\[ \text{such that } 0 \leq c_1g(n) \leq f(n) \leq c_2g(n) \text{ for all } n \geq n_0. \} \]

More succinctly: \[ \Theta(g(n)) = O(g(n)) \cap \Omega(g(n)) \]

5 Anonymous functions

These notations officially refer to sets of functions, but it’s often useful to use them in larger arithmetic expressions.

\[ T(n) = O(n^2) \] means:
\[ T(n) \in O(n^2) \]

\[ T(n) = 2n^2 + O(n) \] means:
[There exists \( f(n) \in O(n) \) with \( T(n) = 2n^2 + f(n) \).]
6 Little-\(o\)

\( o(g(n)) = \{ f(n) \mid \text{for any positive constant } c, \)
\( \text{there exists a constant } n_0 \text{ such that } 0 \leq f(n) \leq cg(n) \)
\( \text{for all } n \geq n_0. \} \)

This indicates a loose bound: \( f(n) = o(g(n)) \) means \( f(n) \) grows strictly slower than \( g(n) \).

7 Little-\(\omega\)

\( \omega(g(n)) = \{ f(n) \mid \text{for any positive constant } c, \)
\( \text{there exists a constant } n_0 \text{ such that } 0 \leq cg(n) \leq f(n) \)
\( \text{for all } n \geq n_0. \} \)

This indicates a loose bound: \( f(n) = \omega(g(n)) \) means \( f(n) \) grows strictly faster than \( g(n) \).

8 Little-\(\theta\)?

9 Limits

For functions that are eventually positive, we can compare asymptotic growth rates using limits.

Let \( L = \lim_{n \to \infty} \frac{f(n)}{g(n)} \), if that limit exists.

Then \( f(n) \) is in . . .

<table>
<thead>
<tr>
<th>( L )</th>
<th>( \omega(g(n))? )</th>
<th>( \Omega(g(n))? )</th>
<th>( \Theta(g(n))? )</th>
<th>( O(g(n))? )</th>
<th>( o(g(n))? )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( x )</td>
<td></td>
<td></td>
<td></td>
<td>( x )</td>
</tr>
<tr>
<td>(0, ( \infty ))</td>
<td>( x )</td>
<td>( x )</td>
<td>( x )</td>
<td>( x )</td>
<td></td>
</tr>
<tr>
<td>( \infty )</td>
<td>( x )</td>
<td>( x )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10 Informal summary of intuition

- \( f(n) = O(g(n)) \) is like \( a \leq b \).
- \( f(n) = \Omega(g(n)) \) is like \( a \geq b \).
- \( f(n) = \Theta(g(n)) \) is like \( a = b \).
- \( f(n) = o(g(n)) \) is like \( a < b \).
- \( f(n) = \omega(g(n)) \) is like \( a > b \).