

Formalizing "like $f(n)$ " (cont.)

- Proportional means "within a constant factor"
- Examples:
 - $O(2n) = O(n)$, because 2 is a constant factor
 - $\Theta(\log_3(n)) = \Theta(\log_2(n)) = \Theta(\log(n)) = \Theta(\ln(n))$
 - $\log_a(x) = (\log(b) / \log(a)) * \log_b(x)$, and $\log(b) / \log(a)$ is a constant factor
 - $\Omega(\log(n^2)) = \Omega(\log(n))$, because $\log(n^2) = 2\log(n)$
- If something takes constant time, we say it is $O(1)$

Identifying runtimes

- Note: typically care about worst-case performance, so $O(f(n))$ notation is most commonly used outside of academic settings
- For-loops are typically $O(n)$, while-loops vary depending on when the loop is broken
- Nested for-loops are typically n^k where k is the number of nested loops
- Accessing array elements by index is $O(1)$
- Searching for an element in an (unsorted) array is $O(n)$
- Binary search is $O(\log(n))$