9. Data Structures and Algorithms :: Order of Growth Summary

Although we would have to rigorously prove the asymptotic time complexity of an algorithm if we were trying to publish a research paper, informally. we can state the complexity by identifying the dominant term (i.e., the one with the largest order of growth) in our complexity function $f(n)$. Here are some examples:

Complexity	**Example Functions**
$O(1)$	$f(n) = 1$, $f(n) = 5$, $f(n) = 92{,}923{,}929{,}239$
$O(lg\ n)$	$f(n) = lg\ n$, $f(n) = -33\ lg\ n$, $f(n) = 4011\ lg\ 12n$, $f(n) = 5(lg\ n + 21)$
$O(n)$	$f(n) = n$, $f(n) = 7n - 13$, $f(n) = -3345n + 21{,}212{,}132{,}312$
$O(n\ lg\ n)$	$f(n) = n\ lg\ n$, $f(n) = 33n\ lg\ n$, $f(n) = 39{,}343n(lg\ n + 170)$
$O(n^2)$	$f(n) = n^2$, $f(n) = -12n^2$, $f(n) = -14n^2 - 133n - 54$
$O(n^3)$	$f(n) = n^3$, $f(n) = 33n^3$, $f(n) = -14n^3 + 1001n^2 - 133n - 54$
$O(2^n)$	$f(n) = 2^n$, $f(n) = (57)(2^n)$, $f(n) = 2^n + n^3 - 1$, $f(n) = 2^n + n^{99{,}999{,}999{,}999{,}999}$