**Orders of common functions**

Here is a list of classes of functions that are commonly encountered when analyzing the running time of an algorithm. In each case, *c* is a constant and *n* increases without bound. The slower-growing functions are generally listed first. See table of common time complexities for a more comprehensive list.

|  |  |  |
| --- | --- | --- |
| **Notation** | **Name** | **Example** |
|  | [constant](http://en.wikipedia.org/wiki/Constant_time) | Determining if a number is even or odd; using a constant-size [lookup table](http://en.wikipedia.org/wiki/Lookup_table) or [hash table](http://en.wikipedia.org/wiki/Hash_table) |
|  | [logarithmic](http://en.wikipedia.org/wiki/Logarithmic_time) | Finding an item in a sorted array with a [binary search](http://en.wikipedia.org/wiki/Binary_search_algorithm) or a balanced search [tree](http://en.wikipedia.org/wiki/Tree_data_structure) as well as all operations in a [Binomial heap](http://en.wikipedia.org/wiki/Binomial_heap). |
|  | [linear](http://en.wikipedia.org/wiki/Linear_time) | Finding an item in an unsorted list or a malformed tree (worst case) or in an unsorted array; Adding two n-bit integers by [ripple carry](http://en.wikipedia.org/wiki/Ripple_carry_adder). |
|  | [linearithmic](http://en.wikipedia.org/wiki/Linearithmic_time), loglinear, or quasilinear | Performing a [Fast Fourier transform](http://en.wikipedia.org/wiki/Fast_Fourier_transform); [heapsort](http://en.wikipedia.org/wiki/Heapsort), [quicksort](http://en.wikipedia.org/wiki/Quicksort) (best and average case), or [merge sort](http://en.wikipedia.org/wiki/Merge_sort) |
|  | [quadratic](http://en.wikipedia.org/wiki/Quadratic_time) | Multiplying two *n*-digit numbers by a simple algorithm; [bubble sort](http://en.wikipedia.org/wiki/Bubble_sort) (worst case or naive implementation), [Shell sort](http://en.wikipedia.org/wiki/Shell_sort), quicksort (worst case), [selection sort](http://en.wikipedia.org/wiki/Selection_sort) or [insertion sort](http://en.wikipedia.org/wiki/Insertion_sort) |
|  | [polynomial](http://en.wikipedia.org/wiki/Polynomial_time) or algebraic | [Tree-adjoining grammar](http://en.wikipedia.org/wiki/Tree-adjoining_grammar) parsing; maximum [matching](http://en.wikipedia.org/wiki/Matching_(graph_theory)) for [bipartite graphs](http://en.wikipedia.org/wiki/Bipartite_graph) |
|  | [exponential](http://en.wikipedia.org/wiki/Exponential_time) | Finding the (exact) solution to the [travelling salesman problem](http://en.wikipedia.org/wiki/Travelling_salesman_problem) using [dynamic programming](http://en.wikipedia.org/wiki/Dynamic_programming); determining if two logical statements are equivalent using brute-force search |

Graphs: <http://science.slc.edu/~jmarshall/courses/2002/spring/cs50/BigO/index.html>

# Java standard data structures Big O notation

