BIG O

UPPER & LOWER BOUNDS

TIME COMPLEXITY	MAX N (LOW END)	MAX N (HIGH END)
$O(n!)$	11	9
$O(2^n)$	27	20
$O(n^3)$	584	125
$O(n^2)$	14,142	1,414
$O(n \log n)$	8,677,239	118,649
$O(n)$	200,000,000	2,000,000
$O(\log n)$	Super high	Super high
$O(1)$	Infinite	Infinite

COMMON COMPLEXITY CLASSES

COMPLEXITY CLASS	NOTATION	DESCRIPTION	EXAMPLES OF FUNCTIONS IN THE CLASS
Constant	$O(1)$	The growth rate does not depend on the input.[1]	$18, 3 * 3 * 3, 10!$ $\min(n, 10)$
Logarithmic	$O(\log n)$	Very slow growth rate. When the input doubles, the value only increases by a constant.	$\log_2(n), \log_3(n)$ $6*\log_2(2n+1)$
Linear	$O(n)$	When the input doubles, the value at most doubles.	$n, n/10, 3n + 5$ $n + \log_2(n)$
Linearithmic	$O(n \log n)$	Slightly faster growth than linear. When the input doubles, the value grows by a bit more than double.	$7n * 3\log_2(n)$ $n * \log_2(n) + n$
Quadratic	$O(n^2)$	When the input doubles, the value at most quadruples.	$n(n-1)/2$ $1 + 2 + 3 + ... + n$
Cubic	$O(n^3)$	When the input doubles, the value grows at most eight-fold.	$n^3 + n^2*\log_2(n)$
Exponential (with base 2)	$O(2^n)$	Extremely fast growth rate. When the input grows by one, the value can double.	$2^n, 2^{(n+1)}, 2^n + n^2$
Exponential (with base 3)	$O(3^n)$	Extremely fast growth rate. When the input grows by one, the value can triple.	$3^n, 3^{(n+4)}, 3^n + 2^n$
Factorial	$O(n!)$	Even faster growth rate. When the input grows by one, the value gets multiplied by a factor that increases each time.	$n! + n!$ $1 * 2 * 3 * ... * n$

1 Even though we learned in Rule 1 to get rid of constants, the constant complexity class is the exception as it is denoted by $O(1)$. It contains all functions where the value does not depend on the input, like $f(n) = 5$. It also contains functions where the value is below some constant for any input, like $f(n) = \min(n, 10)$. This function is never bigger than 10, so $\min(n, 10) = O(1)$.