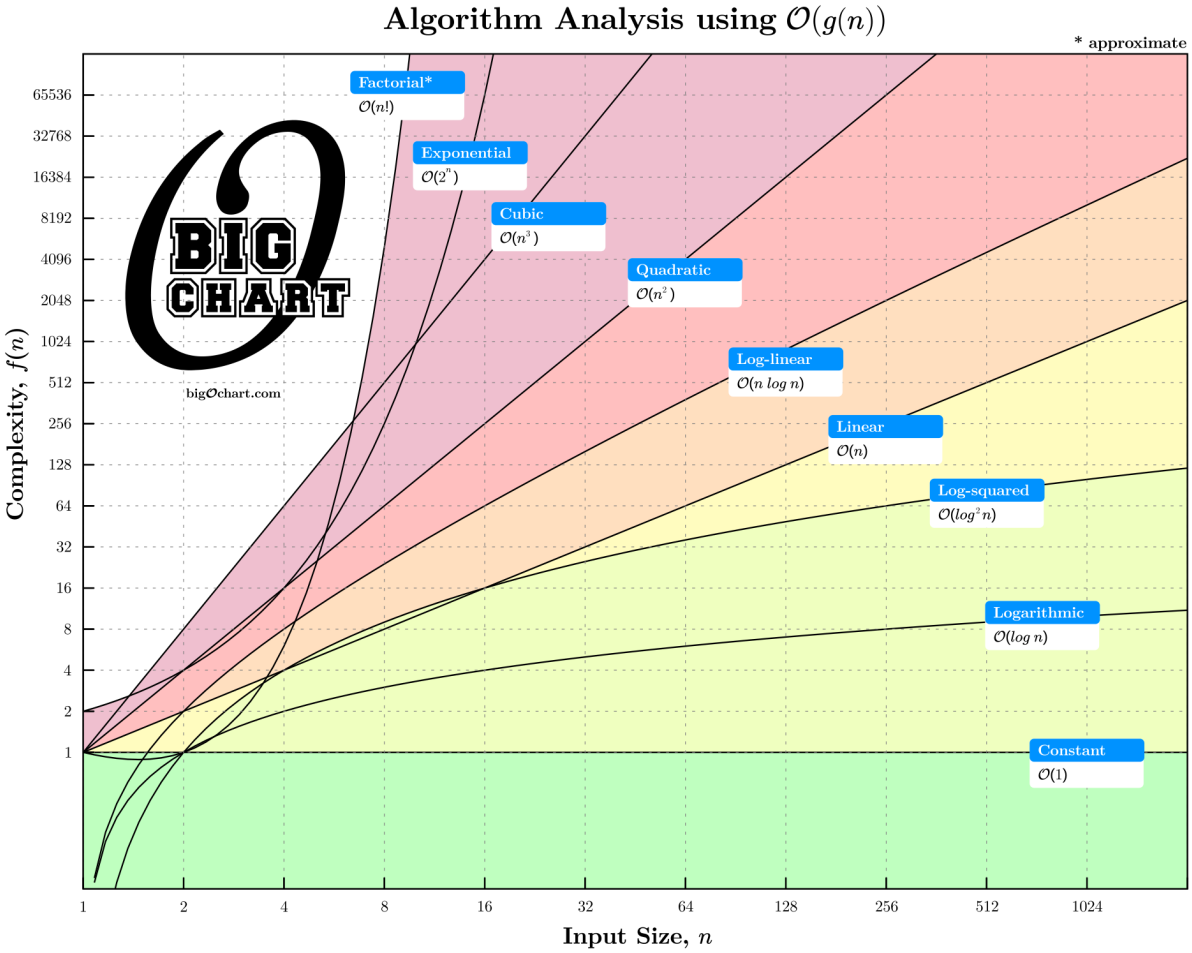

Figure 1: $\mathcal{O}$-notation, a.k.a. "Big-O", is used to express the asymptotic upper bound on $f(n)$ by some constant multiple of $g(n)$, written as $f(n) = \mathcal{O}(g(n))$. This upper bound represents the growth of the worst case running time or space consumption and makes no claims regarding tightness of fit. The shaded area underneath each function depicts the absence of an asymptotic lower bound associated with $\mathcal{O}$-notation. See Table 1 for a partial numerical analysis.

№	Name	$\mathcal{O}(g(n))$	Complexity, $f(n)$						
9	Factorial	$\mathcal{O}(n!)$	2.09×10^{13}	2.63×10^{35}	1.27×10^{89}	3.86×10^{215}			
8	Exponential	$\mathcal{O}(2^n)$	65,536	4.29×10^{9}	1.84×10^{19}	3.40×10^{38}	1.16×10^{77}	1.34×10^{154}	
7	Cubic	$\mathcal{O}(n^3)$	4,096	32,768	262,144	2,097,152	1.68×10^{7}	1.34×10^{8}	1.07×10^{9}
6	Quadratic	$\mathcal{O}(n^2)$	256	1,024	4,096	16,384	65,536	262,144	1,048,576
5	Log-linear	$\mathcal{O}(n \log n)$	64	160	384	896	2,048	4,608	10,240
4	Linear	$\mathcal{O}(n)$	16	32	64	128	256	512	1,024
3	Log-squared	$\mathcal{O}(\log^2 n)$	16	25	36	49	64	81	100
2	Logarithmic	$\mathcal{O}(\log n)$	4	5	6	7	8	9	10
1	Constant	$\mathcal{O}(1)$	1	1	1	1	1	1	1
	Input Size, n		**16**	**32**	**64**	**128**	**256**	**512**	**1,024**

Table 1: Growth rates for algorithms with common complexities.