Math

$log_k(x)$ is the number of times we have to divide x by k to get to 1.

$$log_k(a\,b) = log_k(a) + log_k(b)$$

$$log_k(x^n) = n \cdot log_k(x)$$

$$log_k(\frac{a}{b}) = log_k(a) - log_k(b)$$

$$log_u(x) = \frac{log_k(x)}{log_k(u)}$$

The natural logarithm $ln(x)$ of a number x is a logarithm whose base is $e \approx 2.71828$.
The number of digits of an integer x in base b is $\lfloor log_b(x) + 1 \rfloor$.

Time complexity

We can use time complexity to estimate if a particular algorithm is going to be efficient enough.

Assuming time limit is 1 second:

Input size	Required time complexity
$n \leq 10$	$O(n!)$
$n \leq 20$	$O(2^n)$
$n \leq 500$	$O(n^3)$
$n \leq 5000$	$O(n^2)$
$n \leq 10^6$	$O(n\,log\,n)$
$n \geq 10^6$	$O(log\,n)$

It is important to remember that time complexity is only an estimate.