



Math for the people, by the people.

time complexity

Canonical name	TimeComplexity
Date of creation	2013-03-22 11:47:39
Last modified on	2013-03-22 11:47:39
Owner	akrowne (2)
Last modified by	akrowne (2)
Numerical id	10
Author	akrowne (2)
Entry type	Definition
Classification	msc 68Q15
Defines	polynomial time
Defines	polynomial-time
Defines	exponential time
Defines	exponential-time
Defines	complexity classes
Defines	meta-complexity classes

1 Time Complexity

Time complexity refers to a function describing how much time it will take an algorithm to execute, based on the parameters of its input. The exact value of this function is usually ignored in favour of its order, expressed in the so-called Big-O notation (this is based on the limit of the time complexity function as the values of its parameters increase without bound.)

Complexity classes are equivalence classes of time complexities which are “equal” in Big-O notation. Further, there are meta-complexity classes¹ of time complexities which have Big-O expressions that differ only by some specific parameter. For instance, $O(n^2)$ and $O(n^3)$ are both *polynomial time* complexity classes, similarly $O(2^n)$ and $O(3^n)$ are *exponential time* complexity classes.

2 Example

All comparison-based sorting has time complexity no better than $O(n \log n)$, where n is the number of elements to be sorted. The exact expression for time complexity of a particular sorting algorithm may be something like $T(n) = an \log n + b$, with a and b constants, which still is order $O(n \log n)$.

¹This term has been invented for use in this entry.