

A problem is called an intractable problem if the optimal algorithm for the problem cannot solve all instances of the problem in polynomial time. These types of problems can be solved in theory, but in practice they take too long for their solutions to be useful. Intractable problems are sometimes called computationally intractable, computationally complex, or computationally hard. An intractable problem may be exponential [$O(k^n)$], factorial [$O(n!)$], or super exponential [$O(n^n)$].

constant	$O(1)$
logarithmic	$O(\log n)$
linear	$O(n)$
n-log-n	$O(n \times \log n)$
quadratic	$O(n^2)$
cubic	$O(n^3)$
exponential	$O(k^n)$, e.g. $O(2^n)$
factorial	$O(n!)$
super-exponential	e.g. $O(n^n)$

12.9 P=NP?—The Million Dollar Question

It is discussed that if a problem can be solved by a deterministic Turing machine in polynomial time, it is called P class problem. If a problem can be solved by a deterministic Turing machine in exponential time, it is called E class problem. The time complexity of E class problems are of $O(2^n)$ or $O(n^n)$. If $n = 1, 2, 3, \dots$, then E class problem becomes P class problem. Thus, it can be said that $P \subset E$. On the other hand, if a problem can be solved by a non-deterministic Turing machine in polynomial time, it is called NP class problem. In the chapter ‘Variation of Turing machine’, the conversion process of a non-deterministic Turing machine to a deterministic Turing machine is discussed. There, we have to construct a computational tree consisting of the branches for accepting a string by a non-deterministic Turing machine. Then the tree is traversed by BFS. Traversing this BFS tree means visiting all possible computations of the NTM in the order of increasing length. It stops when it finds an accepting one. Therefore, the length of an accepting computation of the DTM is, in general, exponential in the length of the shortest accepting computation of the NTM. This means that all problems in class NP belong to class E, and they can be solved by a DTM in an exponential time.

It is mentioned that $P \subset E$. Now, NP also belongs to class E. Now, can it be told that $P = NP$ or $P \neq NP$? It is one of the unsolved problems in computer science. NP class problems are polynomial time verifiable. Whether S is a solution or not for an NP problem PROB can be verified easily (in polynomial time). But whether the problem PROB can be solved as easy as verification is the basis for $P = NP$? As an example, consider the subset sum problem. The problem is as follows: is there a non-empty subset from a set of integer numbers whose sum is zero? Let a set of integer numbers be $\{1, 2, -3, 4, -5, 7, 1\}$. Does a subset of this set produce 0? The answer is affirmative as a subset $\{1, -3, -5, 7\}$ produces 0. Thus, the answer is easily verified but does there exist an algorithm to find the answer (construction of the subset) in polynomial time? Till now no such algorithm is found, though there exists an algorithm which can find the answer in $O(2^n)$ complexity, and thus NP.