reflection 7

Video 11 and 12 introduced three classes of questions: P, NP and Exp, while it is debatable whether P = NP. They are useful tools for us to estimate the running time of algorithms.

P refers to questions that take polynomial time to solve, such as questions with a running time of O(n), O(n log n), O(n^2), O(n^k). The letter “P” stands for polynomial complexity (n^k, k is a constant - see the table on the right side, which is from lecture note 12), but P also includes less complicated problems that take less running time.

TABLE 1 Commonly Used Terminology for the Complexity of Algorithms.

Complexity	*Terminology*
$\Theta(1)$	Constant complexity
$\Theta(\log n)$	Logarithmic complexity
$\Theta(n)$	Linear complexity
$\Theta(n \log n)$	Linearithmic complexity
$\Theta(n^b)$	Polynomial complexity
$\Theta(b^n)$, where $b > 1$	Exponential complexity
$\Theta(n!)$	Factorial complexity

Exp stands for questions that takes exponential running time, such as O(2^n), O(3^n), O(2^n * n^k), or more. P is a subset of Exp, meaning that Exp also include questions that take less time to solve, rather than the opposite of P. I think people have to make “whether the running time is exponential or not” a turning point because it is a line between “ever solvable” and “not solvable.” Once the running time is exponential (k^n) or more (like factorial), the running time can exceed years (see the table below, from lecture note 12).

TABLE 2 The Computer Time Used by Algorithms.

Problem Size	*Bit Operations Used*					
n	$\log n$	n	$n \log n$	n^2	2^n	$n!$
10	3×10^{-11} s	10^{-10} s	3×10^{-10} s	10^{-9} s	10^{-8} s	3×10^{-7} s
10^2	7×10^{-11} s	10^{-9} s	7×10^{-9} s	10^{-7} s	4×10^{11} yr	*
10^3	1.0×10^{-10} s	10^{-8} s	1×10^{-7} s	10^{-5} s	*	*
10^4	1.3×10^{-10} s	10^{-7} s	1×10^{-6} s	10^{-3} s	*	*
10^5	1.7×10^{-10} s	10^{-6} s	2×10^{-5} s	0.1 s	*	*
10^6	2×10^{-10} s	10^{-5} s	2×10^{-4} s	0.17 min	*	*

However, NP is an nuanced, mysterious, and controversial concepts. “NP” stands for “non-deterministic” polynomial time, meaning that people don’t know if it is ever possible to solve the problem. However, there seems to be some hope since the time to **verify** the answer is still polynomial. It’s like students can understand the correct answers to math questions, but it’s unclear if they can come up with their original solutions independently.

The most difficult questions in NP are called “NP-complete,” such as the SAT and 3-colorability of graphs. It is unclear whether P = NP. If we can prove one NP-complete question is P, then it means NP = P – if the most difficult question in NP take polynomial time to solve, then the remaining easier questions will just take less time. However, it is also possible that P ≠ NP. If P ≠ NP, then NP-complete questions will not have polynomial running time.