EXPTIME

"EXP" redirects here. For other uses, see Exp.

In computational complexity theory, the complexity class **EXPTIME** (sometimes called **EXP** or **DEXPTIME**) is the set of all decision problems that have **exponential runtime**, i.e., that are solvable by a deterministic Turing machine in $O(2^{p(n)})$ time, where $p(n)$ is a polynomial function of n.

In terms of DTIME,

$$\text{EXPTIME} = \bigcup_{k \in \mathbb{N}} \text{DTIME}\left(2^{n^k}\right).$$

We know

$$\text{P} \subseteq \text{NP} \subseteq \text{PSPACE} \subseteq \text{EXPTIME} \subseteq \text{NEXPTIME} \subseteq \text{EXPSPACE}$$

and also, by the time hierarchy theorem and the space hierarchy theorem, that

$$\text{P} \subsetneq \text{EXPTIME}, \text{ NP} \subsetneq \text{NEXPTIME and PSPACE} \subsetneq \text{EXPSPACE}$$

so at least one of the first three inclusions and at least one of the last three inclusions must be proper, but it is not known which ones are. Most experts believe all the inclusions are proper. It's also known that if P = NP, then EXPTIME = NEXPTIME, the class of problems solvable in exponential time by a nondeterministic Turing machine.[1] More precisely, EXPTIME ≠ NEXPTIME if and only if there exist sparse languages in **NP** that are not in **P**.[2]

EXPTIME can also be reformulated as the space class APSPACE, the problems that can be solved by an alternating Turing machine in polynomial space. This is one way to see that PSPACE ⊆ EXPTIME, since an alternating Turing machine is at least as powerful as a deterministic Turing machine.[3]

EXPTIME is one class in an exponential hierarchy of complexity classes with increasingly more complex oracles or quantifier alternations. The class 2-EXPTIME is defined similarly to EXPTIME but with a doubly exponential time bound $2^{2^{p(n)}}$. This can be generalized to higher and higher time bounds.

1 EXPTIME-complete

A decision problem is EXPTIME-complete if it is in EXPTIME, and every problem in EXPTIME has a polynomial-time many-one reduction to it. In other words, there is a polynomial-time algorithm that transforms instances of one to instances of the other with the same answer. Problems that are EXPTIME-complete might be thought of as the hardest problems in EXPTIME. Notice that although we don't know if NP is equal to P or not, we do know that EXPTIME-complete problems are not in P; it has been proven that these problems cannot be solved in polynomial time, by the time hierarchy theorem.

In computability theory, one of the basic undecidable problems is the halting problem: deciding whether a deterministic Turing machine (DTM) halts. One of the most fundamental EXPTIME-complete problems is a simpler version of this, which asks if a DTM halts in at most k steps. It is in EXPTIME because a trivial simulation requires $O(k)$ time, and the input k is encoded using $O(\log k)$ bits which causes exponential number of simulations. It is EXPTIME-complete because, roughly speaking, we can use it to determine if a machine solving an EXPTIME problem accepts in an exponential number of steps; it will not use more.[4] The same problem with the number of steps written in unary is P-complete.

Other examples of EXPTIME-complete problems include the problem of evaluating a position in generalized chess,[5] checkers,[6] or Go (with Japanese ko rules).[7] These games have a chance of being EXPTIME-complete because games can last for a number of moves that is exponential in the size of the board. In the Go example, the Japanese ko rule is sufficiently intractable to imply EXPTIME-completeness, but it is not known if the more tractable American or Chinese rules for the game are EXPTIME-complete.

By contrast, generalized games that can last for a number of moves that is polynomial in the size of the board are often PSPACE-complete. The same is true of exponentially long games in which non-repetition is automatic.

Another set of important EXPTIME-complete problems relates to succinct circuits. Succinct circuits are simple machines used to describe some graphs in exponentially less space. They accept two vertex numbers as input and output whether there is an edge between them. For many natural P-complete graph problems, where the graph is expressed in a natural representation such as an adjacency

matrix, solving the same problem on a succinct circuit representation is EXPTIME-complete, because the input is exponentially smaller; but this requires nontrivial proof, since succinct circuits can only describe a subclass of graphs.[8]

2 References

[1] Christos Papadimitriou (1994). *Computational Complexity*. Addison-Wesley. ISBN 0-201-53082-1. Section 20.1, page 491.

[2] Juris Hartmanis, Neil Immerman, Vivian Sewelson. Sparse Sets in NP-P: EXPTIME versus NEXPTIME. *Information and Control*, volume 65, issue 2/3, pp.158–181. 1985. At ACM Digital Library

[3] Papadimitriou (1994), section 20.1, corollary 3, page 495.

[4] Du, Ding-Zhu; Ko, Ker-I (2014), *Theory of Computational Complexity*, Wiley Series in Discrete Mathematics and Optimization (2nd ed.), John Wiley & Sons, Proposition 3.30, ISBN 9781118594971.

[5] Aviezri Fraenkel and D. Lichtenstein (1981). "Computing a perfect strategy for n×n chess requires time exponential in n". *J. Comb. Th. A* (31): 199–214. doi:10.1016/0097-3165(81)90016-9.

[6] J. M. Robson (1984). "N by N checkers is Exptime complete". *SIAM Journal on Computing*. **13** (2): 252–267. doi:10.1137/0213018.

[7] J. M. Robson (1983). "The complexity of Go". *Information Processing; Proceedings of IFIP Congress*. pp. 413–417.

[8] Papadimitriou (1994), section 20.1, page 492.

3 Text and image sources, contributors, and licenses

3.1 Text

- **EXPTIME** *Source:* https://en.wikipedia.org/wiki/EXPTIME?oldid=742310238 *Contributors:* AxelBoldt, LC~enwiki, Brion VIBBER, The Anome, Jan Hidders, Arvindn, Edemaine, Marknau, Michael Hardy, Nixdorf, Ixfd64, Charles Matthews, Dcoetzee, Doradus, Jimbreed, Pigsonthewing, Andris, Mboverload, Gdr, Creidieki, Ascánder, DcoetzeeBot~enwiki, EmilJ, Jérôme, Anders Kaseorg, Greg Kuperberg, Anonymous3190, Kenyon, Salix alba, Ucucha, FlaBot, Mathbot, Chobot, RussBot, Trovatore, Cedar101, PhS, SmackBot, Chris the speller, Bluebot, Duncancumming, CRGreathouse, David Eppstein, Rollowicz, Jamelan, Draconx, Plastikspork, SoxBot III, Addbot, DOI bot, LaaknorBot, Jarble, Luckas-bot, AnomieBOT, Citation bot, Miym, Lefschetz, BenzolBot, OlegGerdiy, H3llBot, Widr, Helpful Pixie Bot, GeorgeRu, Spacecowboy420, NatasaVuksanovic123 and Anonymous: 26

3.2 Images

3.3 Content license

- Creative Commons Attribution-Share Alike 3.0