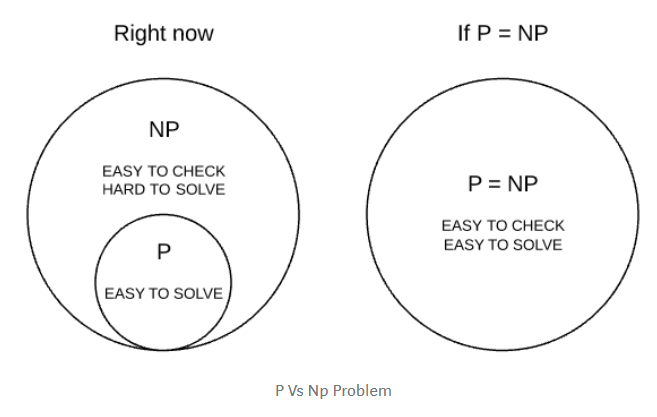
**P versus NP Problem**

* **P (polynomial time):**

P is a class that includes all the problems that can be solved by a reasonably fast program, like multiplication or addition.

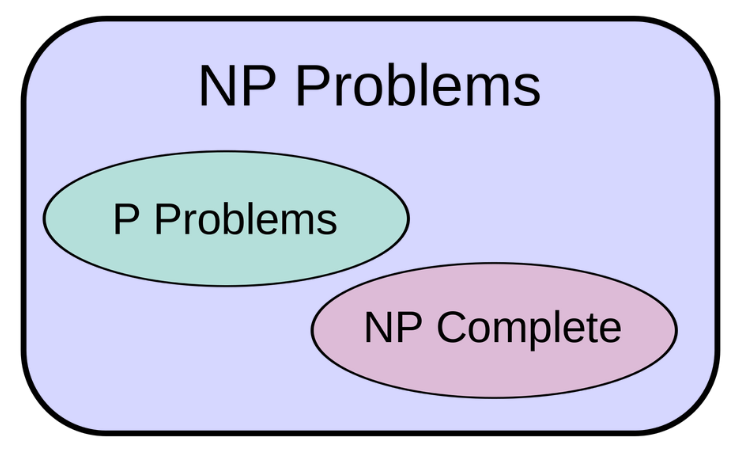


* **NP** (**nondeterministic polynomial time):**

NP is a **complexity *class*** includes all the problems where if you're given a correct solution you can **at least check** it in a reasonable amount of time but computing the correct solution itself is not possible in polynomial time.

**Ex.** **Factorization** can take a long time to solve but if we already know the factors then ,we can easily check it for mistakes.

NP contains lots of important problems **like** vehicle routing, scheduling, circuit design, and databases.



Sometimes we get lucky and find that an NP problem is actually a part of P and we'd have our fast program. But, for a lot of them that didn't seem to be happening.

* **NP-hard:**

A problem is **NP-hard** if an algorithm for solving it can be translated into one for solving any NP-problem (nondeterministic polynomial time) problem.

Or

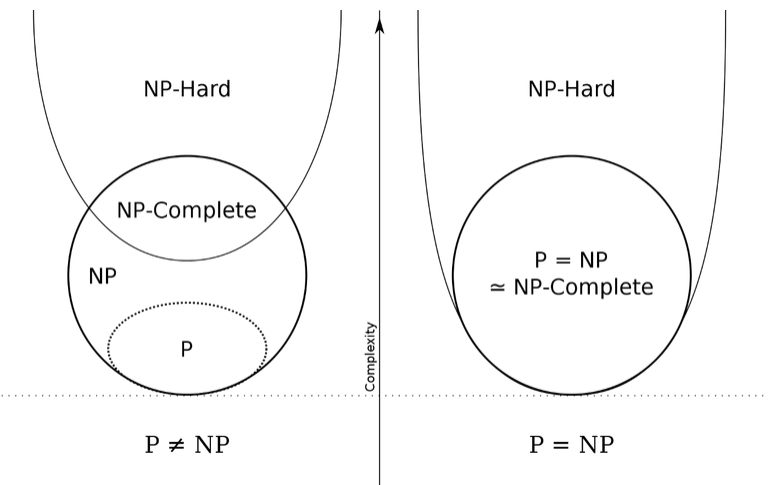
that are **at least** as hard as NP problem hardest part is called "NP-hard".

* **NP-complete:**

A group of problem where if a fast solution to **any of one** the problem is found we can solve a group of problem in same set of complexity with ease

Or

NP-complete means that these problems include all the really **hard parts** of every NP problem.



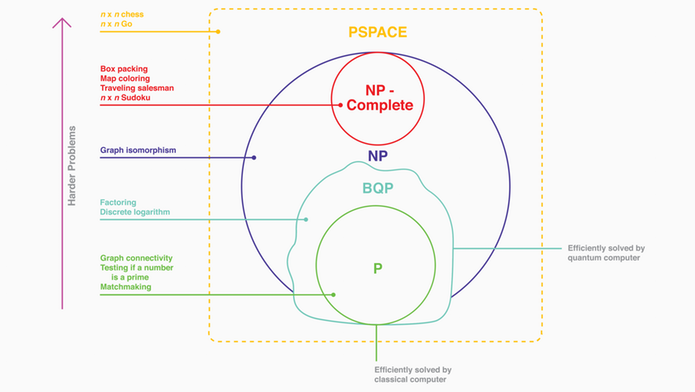
# **The Computational Complexity:**

Theoretically, a fast program for solving any NP-complete problem could be used to solve every problem in NP. The whole class would instantly collapse.

**Ex.**

**Sudoku** is hard because it involves, literally, the same NP-complete task that makes **protein folding** hard.

Therefore, solving the NP-complete task of Sudoku in polynomial time would solve the fast protein folding which in turn would help us cure cancer.



* **"P-SPACE":**

the class of problems that can be solved given **unlimited time**, but using only a polynomial amount of space for memory.

* **“BPP”:**

The class of problems that can be solved **probabilistically** in polynomial time.

* **Conclusion:**

*"****If P were equal to NP, then the world would be a profoundly different place than we usually assume it to be. There would be no special value in "creative leaps", no fundamental gap between solving a problem and recognizing the solution once it's found. Everyone who could appreciate a symphony would be Mozart; Everyone who could follow a set-by-step argument would be Gauss****."*

By Scott Aaronson, a complexity researcher at MIT