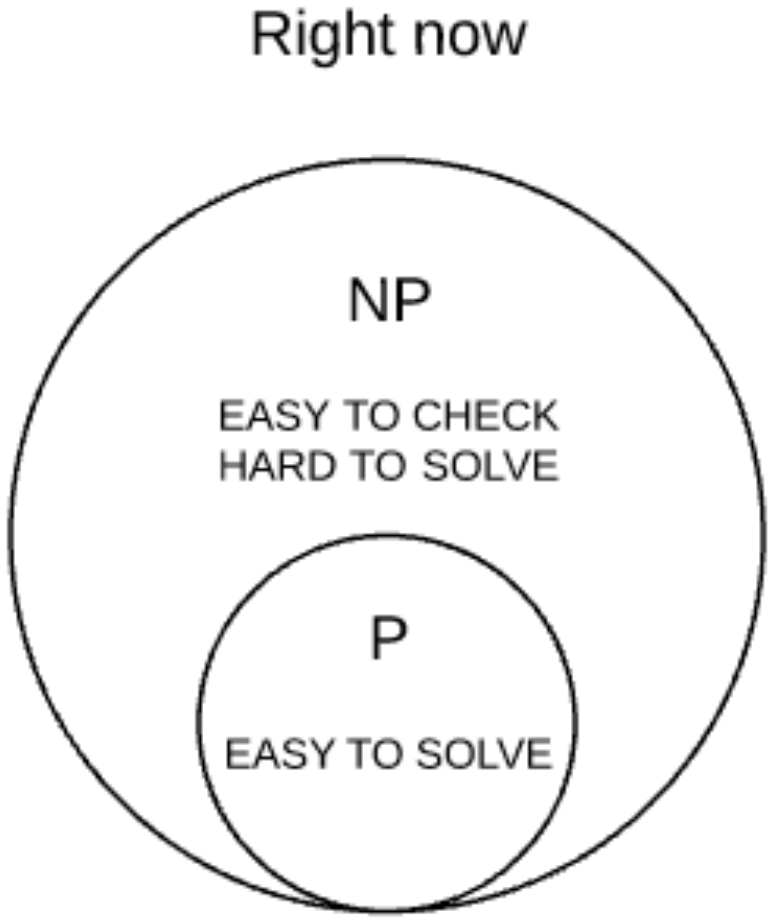
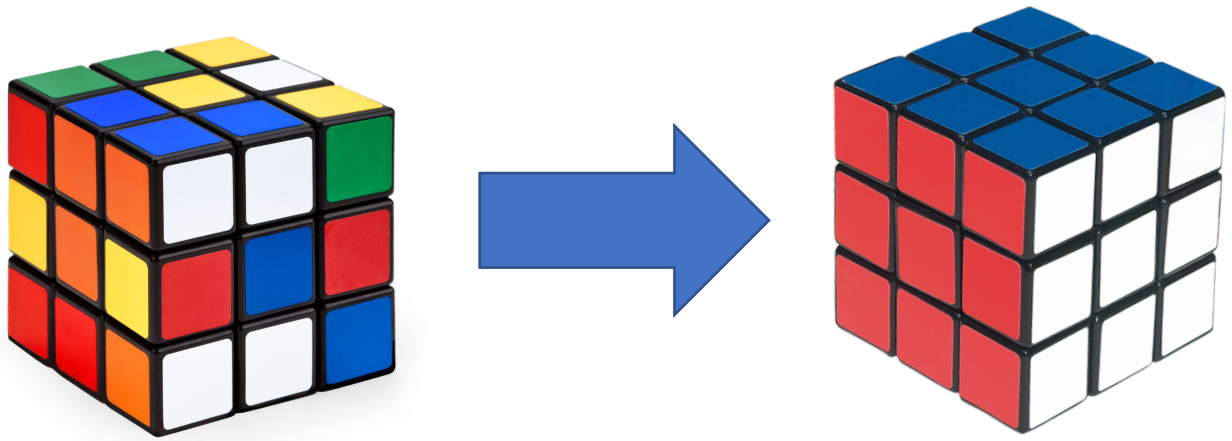
P Versus NP Complexity Theory

- Anuja Nagare

Biggest Unsolved problem in CS





Number Scrabble

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

- 2 Player Game
  - Each player can select a number

- Aim :
  - Pick 3 numbers that add up to 15

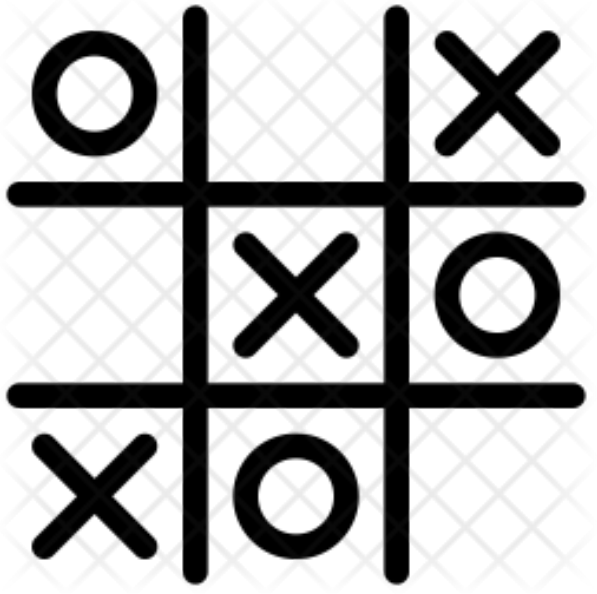
Magic Square

2	**7**	**6**
9	**5**	**1**
4	**3**	**8**

- Each row adds up to 15 !
- Each diagonal adds up to 15 !
- Each column adds up to 15 !

~ TIC – TAC - TOE

2	7	6
9	5	1
4	3	8

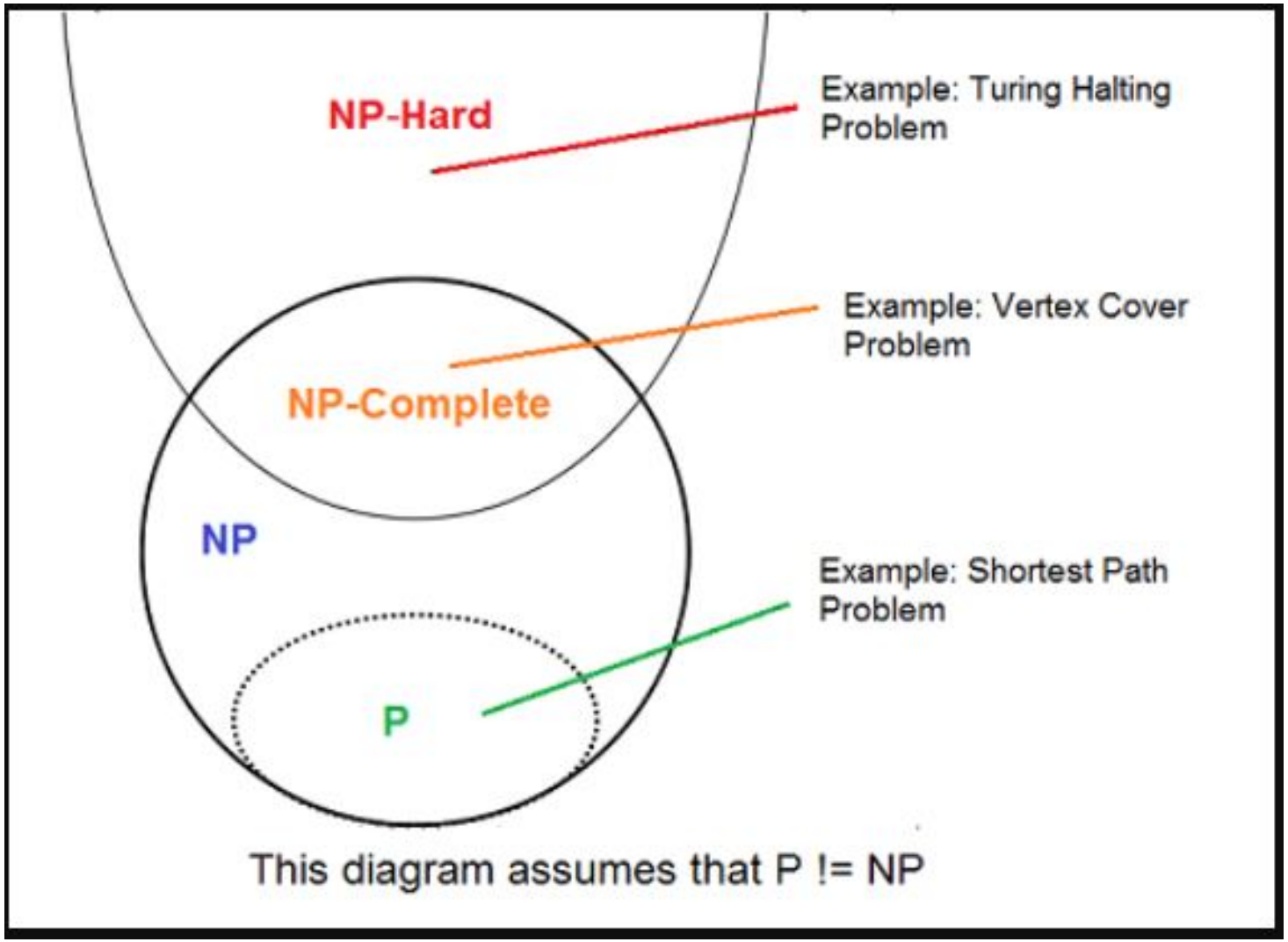


If 2 problems can be reduced down to same problem

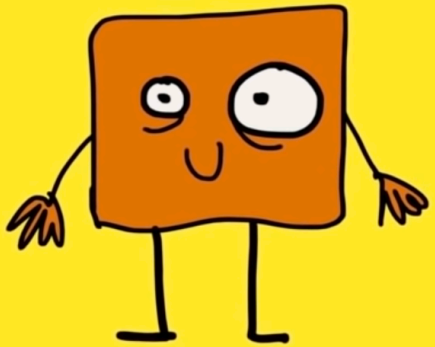
→ then they can be solved by same strategy !!

strategies => mechanical procedures => **Algorithms**!!

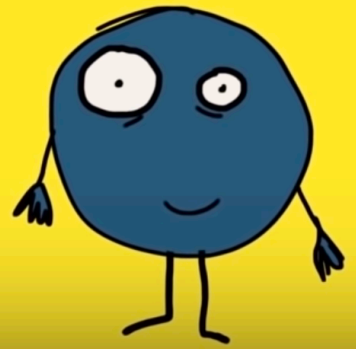
Computational complexity



Matrix Multiplication



Sorting



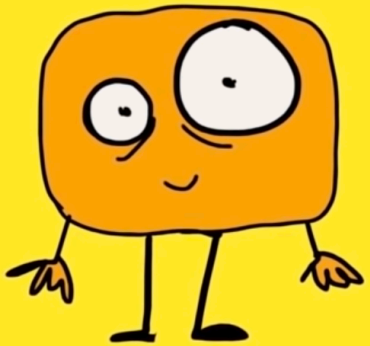
P

Polynomial

n^2 n^5

↑ Number of inputs

Matching



P

- Example :
  - Sort 5 numbers lowest to highest
  - Sort 10,000 number lowest to highest !!

- All Problems that a computer can solve in a reasonable amount of time if “P”

NP

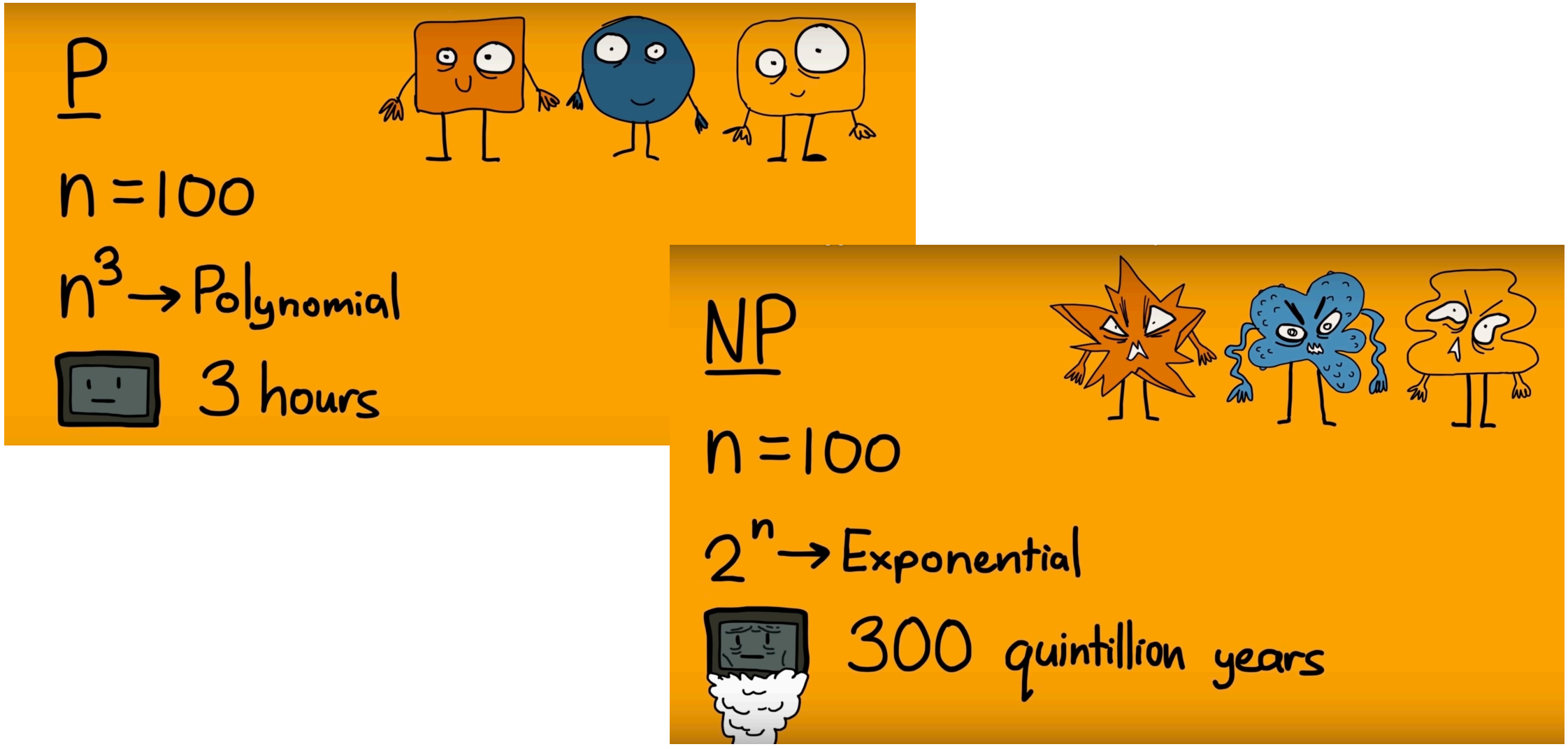
SAT

Travelling Salesman

Problems that can be verified in polynomial time, but may take an exponential number of steps to solve.

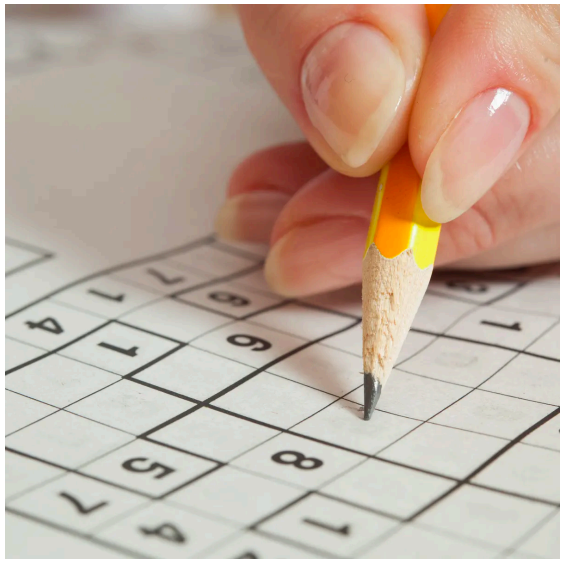
3-Coloring

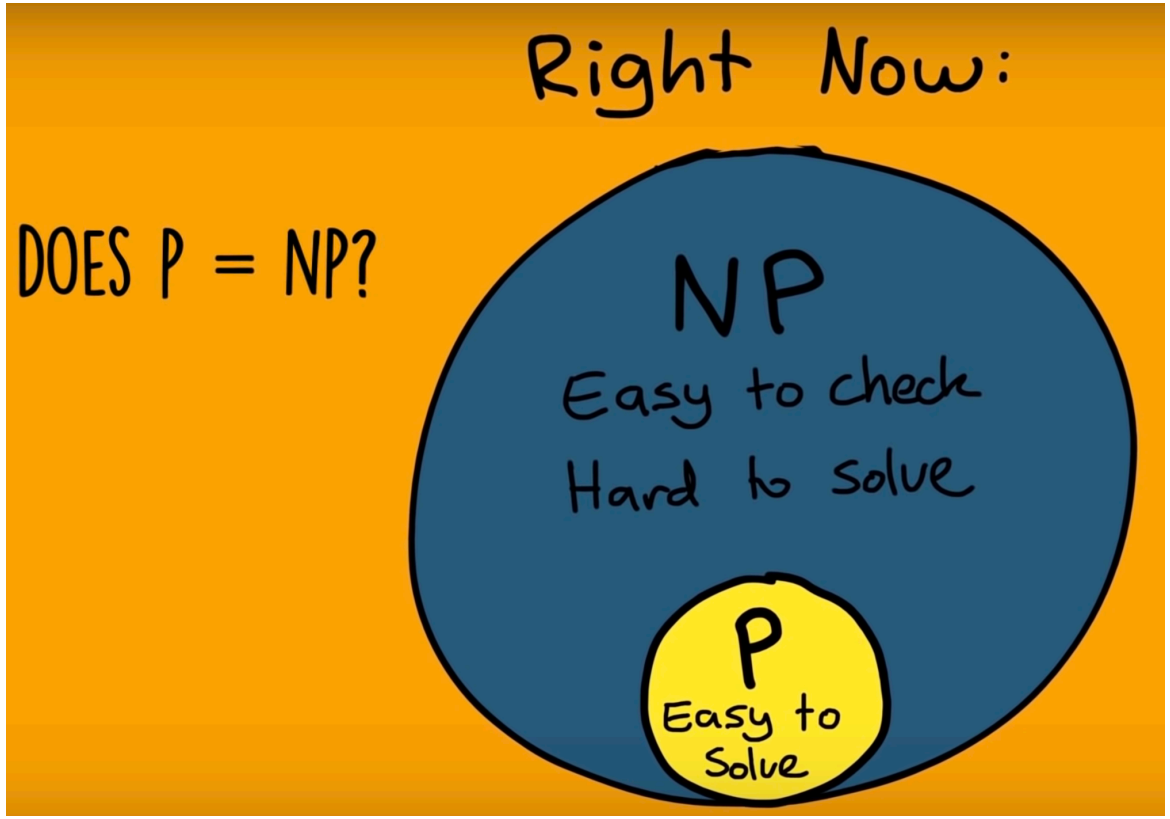
P Vs NP


P
n=100
n^3 → Polynomial
3 hours
NP
n=100
2^n → Exponential
300 quintillion years

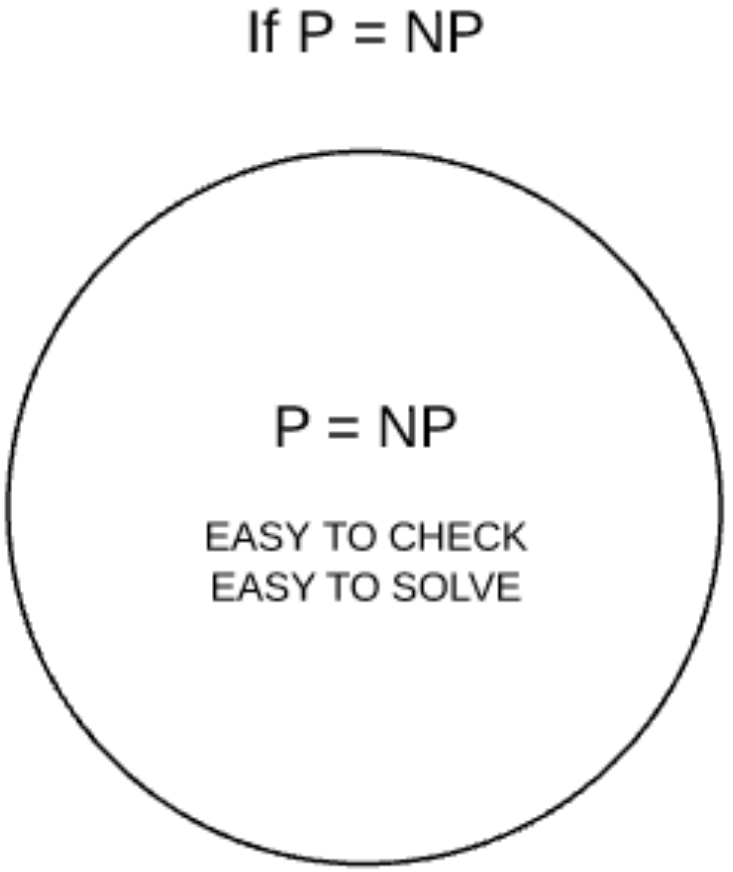
NP

- Relatively short time to verify
- Example : Sudoku
  - Its difficult to solve
  - Its easy to verify if the solution is correct

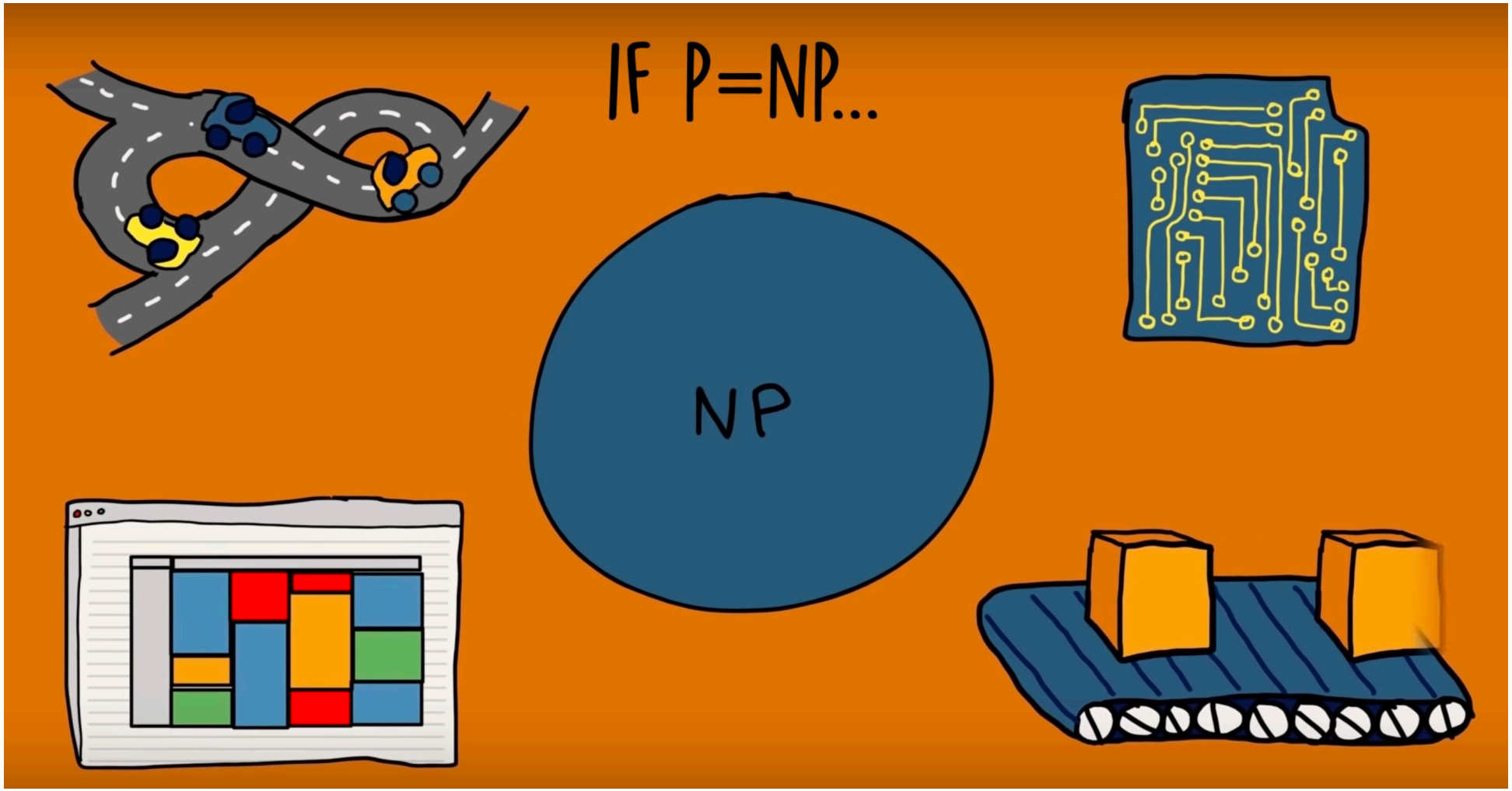


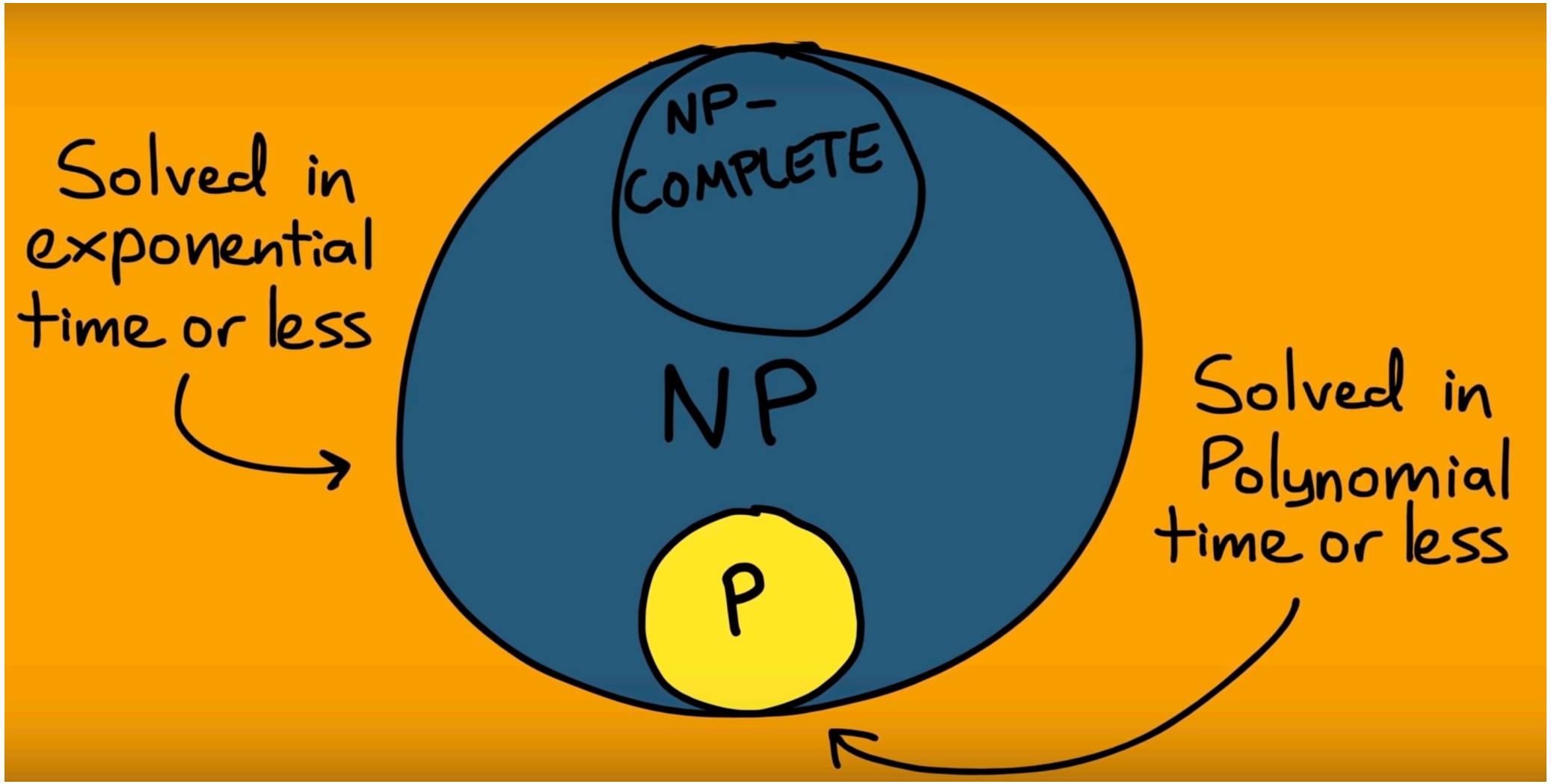


Are questions that are verified in polynomial time also solved in polynomial time ??



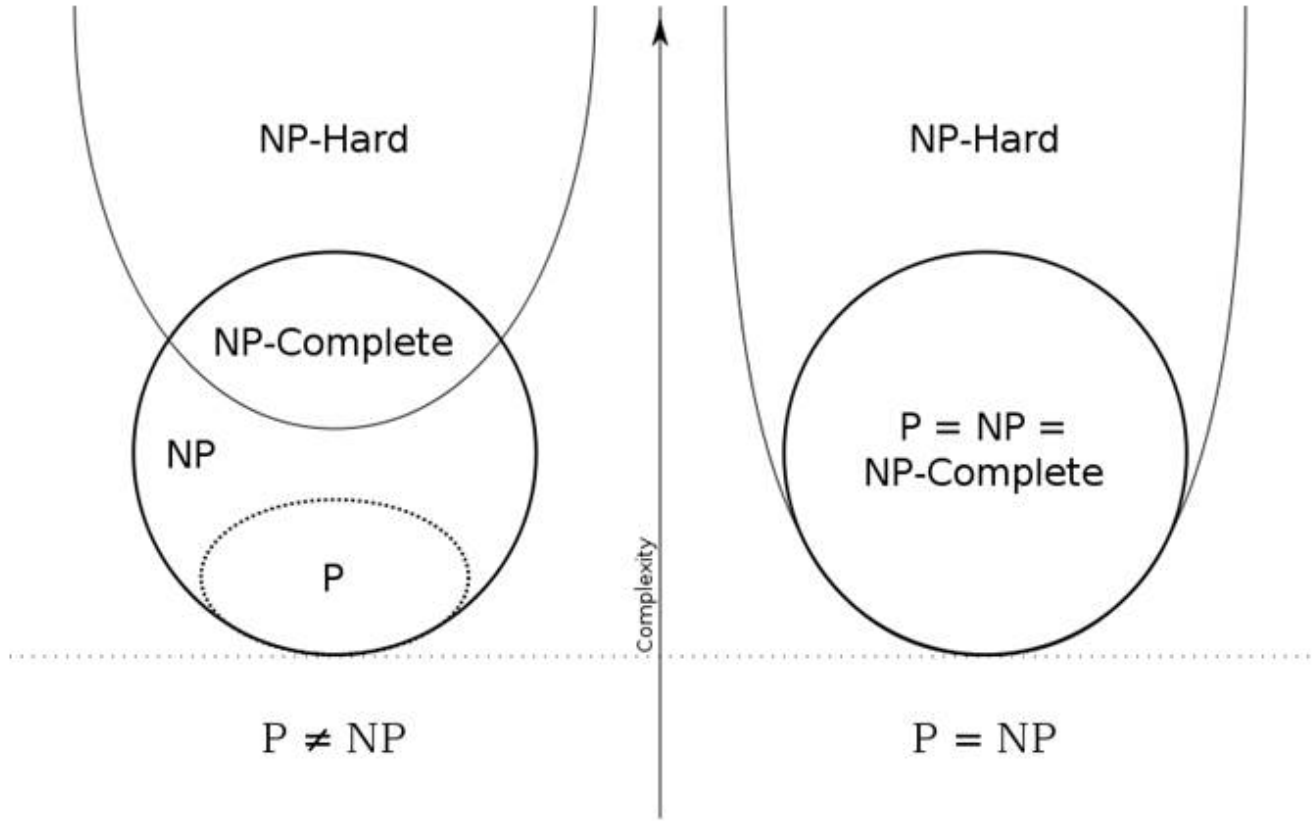
What if !!




Solved in exponential time or less
NP-COMPLETE
NP
P
Solved in Polynomial time or less

P Versus NP Complexity Theory

The **P versus NP problem** is a major unsolved problem in computer science. Informally, it asks whether every problem whose solution can be quickly verified by a computer can also be quickly solved by a computer.



P: Polynomial Time - Algorithm running time is upper bounded by polynomial expression in the size of the input for the algorithm … e.g. it's "feasible", "efficient", "fast", such as Quicksort and all basic arithmetic operations

NP: Can only be solved in Non-deterministic Polynomial–time, such as determining whether two graphs can be drawn identically
NP-Complete: No fast solution is known, such as the Knapsack problem
NP-Hard: At least as hard to solve as the hardest problems in NP – an example is the Traveling Salesman Problem