Can we design an algorithm for any given problem? **NO**

Eg.: Given a number, find the factors (discrete problem)
- Given two points in a room, find a path to move an object between the two points.

- Given a C program, and an input \( x \), what is the output if \( x \) on \( x \)？
  \[ x \rightarrow A \rightarrow \text{Yes/No} \]
There exist "undecidable" problems.

Can we design "efficient" algorithm for these problems for which we can design

\[ n^n \ldots 2^n \]

polynomial time is considered acceptable

\[ n^2, n^3, \ldots, n^{1000} \]

Classification: efficient/not efficient

We can design a polynomial time algorithm:

- I cannot design a polytime algo
- Polytime algorithm is not known to exist
- No one can ever design a polytime algo
- It is "unlikely"
Given a graph $G = (V, E)$ is there a path that visits every vertex exactly once?

No proof that no one can design an efficient algorithm for lower bounds.

There is a class of very interesting natural problems for which polynomial algorithms are not known.

Further, there are other interesting properties:

- If we can solve any of these problems efficiently $\Rightarrow$ efficient solns for the others also "reducibly"
For these problems, the answer can be verified efficiently.

\[
\begin{array}{c|c}
\text{Prover} & \text{Verifier} \\
\hline
\text{fast verification} & \text{using "certificate"} \\
\end{array}
\]

Given a specific graph.

This model is more suitable for decision problems (rather than optimization problems).

- Decision Knapsack: Is there a solution with profit \( \geq P \)?

- What is the min no. of colors required to "legally color a graph" (no two vertices with an edge can have the same color)?

Can we color this graph using 4 colors?
Is the graph 100 colorable

Yes

[Verifier]

produce a coloring using 100 colors

No

→ Is the graph not 100 colorable

The complement of the class NP is called co-NP.

and co-NP may not have an efficient verifier.