L = \{ <R1, R2> \mid \text{R1, R2 are regular expressions over the operations concat, union, * and square and } L(R1) = L(R2) \}
the class $P$
- problems solved in polynomial time.

$$P = \bigcup_{k} \text{TIME}(n^k)$$

Problem: sort a list
algorithm: bubble sort $O(n^2)$
          quick sort $O(n \log n)$
          merge sort $O(n \log n)$
          heap sort $O(n \log n)$
NP: non-deterministic polynomial.
Euler paths

- visit each edge exactly once

Hamiltonian Path
- go through each node exactly once

Currently no one has found an algorithm in P for Hamiltonian paths.

Exponential algorithm
- for each ordering of nodes
  - check if the ordering is a Ham. path.

Non-deterministic Alg.
- for all orderings (at once)
  - check if there is a Ham. Path

\[ \text{\textbf{NP}} \]
\[ P \subseteq NP \]

\[ P \subseteq NP \]

\[ P \neq NP \]

\[ P: \text{ Languages computable by } \]
\[ \text{det. poly. time } T \text{m.} \]

\[ NP: \text{ non-det. poly. time } T \text{m.} \]

**NP-class of languages w/ P time verifiers.**

**Verifier:** given a string, is that string in the language.
\[ \text{HAMPATH} = \{ <G, s, t> \mid G \text{ is } \]
\begin{align*}
&\text{a directed graph with } \\
&\text{a Hamiltonian path from } \\
&\text{node } s \text{ to node } t \} \\
\]

A verifier for a language \( L \) is an algorithm \( V \)

where

\[ L = \{ w \mid V \text{ accepts } <w, c> \text{ for some string } c \} \]