P vs. NP (Overview)

**P:** set of problems solvable in polynomial time

- **SORT**
- **MST**
- **MULT**
- **EP**

**NP:** set of polynomially verifiable problems

- **HAM-PATH:** Determine if \( G \) has a Hamiltonian Path
  - visits all vertices exactly once

**HAM-PATH-VERIFY \((G, p)\):** verify \( p \) is a HP in \( G \)
- follow path, count visits of verts
- make sure each edge exists
- check all counts = 1
- if all checks pass, YES, else NO

\[ \text{time} = O(V^2) \]

**COMPOSITE:** given pos int \( n \), does \( n \) have non-trivial factors \( p \geq \sqrt{n} \)

**COMP-VERIFY \((N, p)\):** size of input: size of most compact representation
- complete \( x = p \cdot q \)
- return \( N = x \)

\[ \text{time} = O(N) \]

\[ \text{size of int } N = \log_2 N \]

**LONG-PATH:** given \( G, k \), determine if \( G \) has a simple path with \( \geq k \) edges

**LONGEST-PATH:** given \( G \), find length of longest simple path

**CIRCUIT-SAT:** given a combinational circuit, is there an input that makes output 1

**NP-complete:** the hardest NP problems

- \( X \) is NP-complete if \( X \in \text{NP} \) and for all other problems \( Y \in \text{NP} \), \( Y \) can be reduced to \( X \)
If we can find a poly-time algorithm for one NPC problem then \( P = NP \)

If we can prove a super-polynomial lower bound for one NP problem then \( P \neq NP \)