**COMPLEXITY CLASSES**

Some problems can be solved via. a simple algorithm

Some have **polynomial O(V2)** worst-case performance

Some have **exponential O(2V)** worst-case performance

Classes of algorithms

* **P** = algorithm can compute answer in polynomial time E.g. sorting algorithms: nlogn, n2
* **NP** = no P algorithm is known for solving the problem

The P and NP classes suggest “level of difficulty”

* **P**… “easy”… has known polynomial-time algorithms
* **NP**… “hard”.. no P algorithm known

Levels of “difficulty”

* **Easy** = have a polynomial-time algorithm (useful in practice)
* **Tractable** = have an algorithm, feasible only for small *N* (small input-data)
* **Intractable** = no tractable algorithm is known (NP-hard)
* **Non-computable** = no algorithm can exist (no solution)

**GRAPH ALGORITHMS**

**Connectivity** = Simple Graphs

**Path Finding** = Simple / directed graphs

**Minimum Spanning Trees** = Weighted graphs

**Shortest Path** = Weighted Graphs

**FINDING A PATH**

Giving a graph, is there a path that leads from Vertex v to Vertex w?