

## P, NP, NP-Complete problems

- Read section 11.3 (pages 401-409)
- When do we call a problem **tractable**?
- What are some problems that are tractable? What are some problems that are intractable?
- What are **decision problems**? How do we represent the class of all tractable decision problems?
- Are the following problems in P or not?
  - Given a graph, determine if the graph is complete.
  - Given a graph, determine if the graph has any triangles in it.
  - Given a graph, determine if we can assign a given number of colors to the vertices so that adjacent vertices have different colors.
  - The traveling salesman problem phrased as a decision problem: Given the city distances and an overall goal C, determine if you can travel through all the cities without ever visiting a city twice except at the end, so that you don't travel a total distance more than C.
  - The Knapsack problem as a decision problem: Given a list of items with values and weights, a knapsack with a given weight limit W and a target value goal V, determine if you can find a set of items that do not exceed the weight limit W but do reach the target value V.
- We represent by NP the class of all decision problems that are polynomial-time-verifiable. (this differs from the book definition but is equivalent). Which of the above problems are NP?
- Open question in Computer Science: Is  $P = NP$ ?
- When do we say that a problem is **polynomially reducible** to another problem?
  - Explain how we can polynomially reduce the problem of finding a Hamiltonian path to that of solving the traveling salesman problem.
- When is a decision problem called **NP-complete**?
  - If a problem  $D_1$  is NP-complete, and it can be polynomially reduced to a problem  $D_2$ , then that problem is also NP-complete.
- Practice problems: 11.3.2, 11.3.4, 11.3.5, 11.3.8, 11.3.9