Exhaustive Search

- Read 3.4, pages 115-120
  - Describe the **Traveling Salesman problem** in terms of a graph.
  - What would the exhaustive-search approach to the TSP entail?
  - What shortcuts could we try to take to the exhaustive-search approach to the TSP?
  - Explain the figure for the number $\frac{1}{2}(n-1)!$ for the number of possible permutations we need to consider for TSP.
  - If $n = 20$, then this formula results in 6.08×10^{16} different permutations. If we can check 10 billion (1×10^{9}) permutations per second, how many days would it take to go through all of them?
  - Describe what the **knapsack problem** is.
  - What would an exhaustive-search approach to the knapsack problem entail?
  - Why can't we simply try to solve the knapsack problem by with an algorithm like: "Start with the largest-value item that fits in the knapsack, put that in, then continue with the largest-value item that fits in the remaining space, and so on"?
  - Describe the **assignment problem**.
  - How can we relate the assignment problem to permutations?
  - Consider the following problem: Peter has 8 hours of work available. He has a number of different projects he can work on. Each would take him a certain amount of time and also provide him with a certain reward. His goal is to maximize his reward. Does this problem fit into one of the three kinds of problems we saw in this section?
  - Consider exercise 3.4.6, describing the partition problem. Describe an exhaustive-search approach, and any optimizations that you can perform for it (if we keep in mind the symmetry in the two sides of an answer).