Questions and Exercises to work out and turn in:

Grading Guidelines:

* A right answer will get full credit when:

1. It is right (worth 25%)
2. It is right **AND** neatly presented making it easy and pleasant to read. (worth an **extra** 15%)
3. There is an **obvious and clear link** between 1) the information provided in the exercise and in class and 2) the final answer. A clear link is built by properly writing, justifying, and documenting an answer (worth an **extra** 60%).
4. Calculation mistakes will be minimally penalized (2 to 5% of full credit) while errors on units will be more heavily penalized.

You are welcome/encouraged to discuss exercises with other students or the instructor. But, ultimately, **personal** writing is expected.

* USE THIS FILE AS THE STARTING DOCUMENT YOU WILL TURN IN. **DO NOT DELETE ANYTHING FROM THIS FILE:** JUST **INSERT** YOUR ANSWERS.
* IF USING HAND WRITING (STRONGLY DISCOURAGED), **USE THIS FILE** BY CREATING SUFFICIENT SPACE AND WRITE IN YOUR ANSWERS.
* FAILING TO FOLLOW TURN IN DIRECTIONS /GUIDELINES WILL COST **A 30% PENALTY.**

Objectives of this assignment:

* to use and manipulate the concepts presented in this module
* to propose and write algorithms in pseudocode
* to analyze the time complexity of algorithms
* to analyze the space complexity of algorithms
* to learn autonomously new concepts

What you need to do:

Answer the questions and/or solve the exercises described below.

Problem (100 points)

The objective is to determine whether the Knapsack problem is NP-complete.

1. (5 points) Research the *Knapsack* problem and **state** it formally
   1. Given a knapsack of weight capacity and items, each item with a weight and a value Determine the number of each item to include in the collection so that the total weight and the value is as large as possible.
2. (10 points) **Which** NP-complete problem among the ones presented in Chapter 34 is the closest to Knapsack (**Hint**: see the map of NP-complete problems shown on the slides and in the textbook). Let us call this problem ***Problem B***.
   1. If **Problem A** is the Knapsack problem listed above, then the closest **Problem B** would be the SUBSET-SUM problem.
   2. The SUBSET-SUM problem is discussed on page 1097 of the textbook and is stated:
3. (5 points) **Describe** an instance for the Knapsack problem.
   1. A set of items with non-negative weights, a maximum weight values , and minimum total value .
4. (5 points) **Cast** the Knapsack problem as a decision problem.
   1. Does there exists a subset of weights with total weight , such that the total value .
5. (5 points) **Describe** an instance for *Problem B*
   1. A set of non-negative integers and total sum
6. (5 points) **Cast** *Problem B* as a decision problem.
   1. Does there exists a subset of numbers where the total sum of
7. (15 points) Propose a **reduction algorithm** to reduce *Problem B* to the Knapsack problem. Ensure that the reduction algorithm runs in proportional time
   1. A reduction algorithm would take the set S in the subset sum problem and use that as S in the knapsack problem. Then, another set would be created with the same entries as S to be used as the set of values V in the knapsack problem. Finally, the sum t in the subset problem would be used as both the maximum weight b and the minimum value goal k. This reduction algorithm will take linear time, since it is simply copying a set and making a few assignments. Linear time is polynomial time, so the reduction algorithm runs in polynomial time.
8. (10 points) Show that the Knapsack problem is in NP
   1. To show that a problem is in NP, it must be shown that the solution to the problem can be checked in polynomial time. The knapsack problem can be checked simply by adding up the weights and the values separately in the set that holds the solution to the problem. Then, compare the sum of the weights to b. If the sum of weights is greater than b, the answer is not a solution. If the sum of the values is less than k, the answer is not a solution. Otherwise, the answer is a solution. This can also be done in linear time, since at most the number of additions performed will be equal to the number of entries in the original set minus 1.
9. (30 points) Show that the answers for both problems are the same
   1. The subset sum problem will return true if there is some sum of the entries in its set that gives the value t, otherwise it will return false. For the knapsack problem, the algorithm will return true if the sum of some of the entries in the set of weights has a weight less than or equal to b, and the corresponding values sum to greater than or equal to k. The reduction algorithm makes the sets the same across all problems, and makes t = k = b. The entries are the same in both sets in the knapsack problem, so the sums for both will end up being the same number, which of course is the same number as the subset sum problem since that set is the same as well. This means that the sum of the weights = the sum of the values, so they are interchangeable. Thus, the sum of the weights must be less than or equal to b, as well as be greater than or equal to k. Since k = b, the sum of the weights must be exactly equal to k, which is also equal to t. In this way, the knapsack problem will only return true if there is some sum of the numbers in the subset sum problem that equals t. Otherwise, it will return false, as will the subset sum problem.
10. (10 points) Is the Knapsack problem NP-complete?
    1. In order to show that a problem is NP-complete, it must be shown that the problem is in NP, and that an already NP-complete problem can be reduced to it. By answering these questions it has been shown that the knapsack problem is in fact NP-complete!

What you need to turn in:

* Electronic copy of this file (including your answers) (standalone). Submit the file as a Microsoft Word or PDF file.
* Recall that answers must be well written, documented, justified, and presented to get full credit.
* How this assignment will be graded:
* A right answer will get full credit when:
* It is right (worth 25%)
* It is right AND neatly presented making it easy and pleasant to read. (worth 15%)
* There is an obvious and clear link between 1) the information provided in the exercise and in class and 2) the final answer. A clear link is built by properly writing, justifying, and documenting an answer (worth 60%).
* Calculation mistakes will be minimally penalized (2 to 5% of full credit) while errors on units will be more heavily penalized.
* You are welcome/encouraged to discuss exercises with other students or the instructor. But, ultimately, personal writing is expected.