**COMP 7270 Assignment 4 No late submissions!**

**Upload your submission well before this deadline. Even if you are a few minutes late, as a result of which Canvas marks your submission late,** **your assignment may not be accepted**.

Instructions:

1. This is an individual assignment. You should do your own work. **Any evidence of copying will result in a zero grade and additional penalties/actions.**
2. Late submissions **will not** be accepted unless prior permission has been granted or there is a valid and verifiable excuse.
3. **Think carefully; formulate your answers, and then write them out concisely** using English, logic, mathematics and pseudocode (no programming language syntax).
4. Algorithms should be provided in numbered pseudocode steps.
5. **Type your answers in this Word document and submit it. If that is not possible, use a word processor to type your answers as much as possible (you may hand-write/draw equations and figures), turn it into a PDF document and upload**.

All questions carry equal weight

# Greedy Algorithm

1. (16.1-2) To prove that the stated approach yields an optimal solution, we have to prove two things: (1) that the choice being made is the greedy choice (this proof will be along the lines of the proof of Theorem 16.1 in the text; but do not copy that proof; your proof will be different, yet similar in structure), and (2) that the resulting solution has optimal substructure.

2. (16.1-4) Suppose that we have a set of activities to schedule among a large number of lecture halls, where any activity can take place in any lecture hall. We wish to schedule all the activities using as few lecture halls as possible. Give an efficient greedy algorithm to determine which activity should use which lecture hall.

3. (16.2-1) Prove that the fractional knapsack problem has the greedy-choice property.

4. (16.2-7) Suppose you are given two sets *A* and *B*, each containing n positive integers. You can choose to reorder each set however you like. After reordering, let *ai* be the *i*-th element of set *A*, and let *bi* be the *i*-th element of set *B*. You then receive a payoff of ../../../../../../../Desktop/Screen%20Shot%202019-04-15%20at%. Give an algorithm that will maximize your payoff. Prove that your algorithm maximizes the payoff, and state its running time.

5. Let's consider a long, straight country road with *n* houses scattered very sparsely along it. We can picture this road as a long line segment with an eastern endpoint and a western endpoint.

Let *di* be the distance of the *i*-th house from the eastern endpoint. We are given as input *di*, for all . Further, let’s suppose that despite the bucolic setting, the residents of all these houses are avid cell phone users. You want to place cell phone base stations at certain points along the road, so that every house is within four miles of one of the base stations. Give an efficient algorithm that achieves this goal, using as few base stations as possible. Identify the complexity of your algorithm.