Institut for Matematik og Datalogi
Syddansk Universitet

Exam Assignment 3
Complexity and Computability — 2018

This is the third of three sets of problems (assignments) which together with the oral exam in June constitute the exam in DM553. This third set of problems may be solved in groups of up to three.

The assignment is due at 10:15 on Friday, June 8. You may write this either in Danish or English. Write your full name (or names if you do it together — up to three people may work together) clearly on the first page of your assignment (on the top, if it's not a cover page). Turn it in as a PDF file via Blackboard through your DM553 course. The assignment hand-in is in the menu for the course and is called "SDU Assignment". Keep the receipt it gives you proving that you turned your assignment in on time. Blackboard will not allow you to turn in an assignment late.

Cheating on this assignment is viewed as cheating on an exam. Do not talk with anyone outside of your group (or David Hammer or Joan Boyar) about the assignment, and do not show your solutions to anyone outside your group. If you have questions about the assignment, come to Joan Boyar or David Hammer.

Assignment 3

Do the following problems. Write clear, complete answers, but not longer than necessary.

1. Consider the APPROX-VERTEX-COVER algorithm on page 1109 of CLRS. Present one graph $G = (V, E)$ with the following property: If the edges are presented in one order, APPROX-VERTEX-COVER finds a minimum vertex cover for G, while if the edges are presented in another order, APPROX-VERTEX-COVER find a vertex cover with size exactly twice the size of a minimum vertex cover. Your graph must have at least ten vertices and be connected. Explain which orderings of the edges give which result.

2. Suppose you have a set of sensors located at various points in a building. These sensors collect data (maybe audio or movement) which should be collected and saved. If all sensors can broadcast their data to those sensors within range, then the data collectors (which we assume are more expensive) only need to be placed with some of the sensors such that each sensor either has a data collector itself or is within range of a sensor with a data collector. Given the knowledge of which sensors are within range of which other sensors (assume that that if sensor A is within range of sensor B, then sensor B is also within range of sensor A), the data collector placement problem is to find a minimum size subset of the sensors which should have data collectors placed with them.

   (a) Show that this data collector placement problem is NP-hard. Reduce from Vertex Cover. Given an instance to Vertex Cover, consider adding a new vertex and two new edges for each original edge in the instance.

   (b) Show how to get an approximation algorithm for the data collector placement problem from the approximation algorithm for set cover. Argue that it works and compute the approximation ratio for the data collector placement problem from the result for set cover. In addition, consider the special case where each sensor is in the range of at most five other sensors, and explain what effect this has on the approximation ratio.

   (c) Consider the problem of checking if a claimed solution to the data collector placement problem really is a solution. You need to check, for every sensor that it is either in the solution or one of the sensors within its range is in the solution.

   Suppose the operations you are allowed to perform are to check if sensor i is in the range of sensor j and to check if a sensor k is in the solution. Assume that there are n sensors.

      i. Suppose your claimed "solution" contains all but three of the sensors. Show how to check if it is actually a solution using $O(n)$ checks.

      ii. Suppose your claimed "solution" only contains three of the sensors. Show how to check if it is actually a solution using $O(n)$ checks.

iii. Use an adversary argument to show that $\Omega(n^2)$ checks must be made in the worst case.

3. Consider algorithms for partitioning a list of $3n$ elements into two sets, one of size $2n$ and one of size n, such that the smallest $2n$ elements are all in the first set, and the largest n elements are in the second set. Suppose the algorithms can be modelled by decision trees which have as their basic operations a comparison operation which can compare either 2 and 3 items in one operation and give the ordering between them (the operation will tell which is largest and which is smallest, and this tells which is middle when there are three items).

   (a) Use an information theoretic argument to prove a lower bound on the number of these comparisons any such algorithm would need to make. How many leaves does your tree have and what is the degree of each node in the tree? (You will need to approximate, but you can still give a lower bound.)

   (b) Give a linear time algorithm for solving this problem.