Institut for Matematik og Datalogi
Syddansk Universitet

DM553 Assignment 3
DM508 Assignment 1

This is your third assignment in DM553 or your first in DM508. The assignment is due at 8:15 on Monday, May 4. You may write this either in Danish or English. Write your full name (or names if you do it in a group of two or three) 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 or DM508 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. If you have questions about the assignment, come to Joan Boyar or Christian Nørskov.

Please note that you must have this assignment approved in order to pass DM553 or DM508. If it is not turned in on time, or if you do not get it approved, it will count as your only retry in the course, and you must have it approved on your only allowed retry for this assignment.

Assignment

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

1. Prove that the following problem, which we will call Gadget–Assembly, is NP-Complete. Gadget–Assembly is a game where the player is given a set of parts (via pictures) and a goal set of gadgets (also via pictures). The player should create each of the specified gadgets, one at a time from the parts, using as few moves as possible. Each gadget contains a power supply (which is one of the parts the player is given) and two other of his/her parts. Combining two parts or a part with something already assembled counts as one move. Taking one part off something

already assembled or partly assembled is free if the result is something the player has previously had in that form.

For example, if the player has parts A (the power supply), B, C, D and E, and must create (A, B, C), (A, B, D), (A, B, E), and (A, C, E), it can do this with a score of 6 (low scores are good) as follows: Combine A and B to get (A, B). Combine (A, B) with C to get (A, B, C). Remove C from (A, B, C) for free to get back to (A, B) which it had before. Combine (A, B) and D to get (A, B, D). Remove D for free. Combine (A, B) and E to get (A, B, E). Remove E for free and remove B for free. Combine A and E to get (A, E). Combine (A, E) with C to get (A, C, E). Note that the order the parts are combined in does not matter, as seen with the last gadget. Also, the gadgets do not need to be created in the order given; it would have been OK to create (A, B, D) before (A, B, C), for example. However, we could not have removed B from (A, B, E) for free, because (A, E) had not existed before.

The Gadget–Assembly problem is, given a set of parts and a goal set of gadgets, is it possible to create the gadgets with a score (number of moves) of at most L?

Part of your proof should be a reduction from the Vertex Cover problem.

2. Suppose that the goal set of gadgets contained gadgets with four parts each (one power supply and three other parts). Is the corresponding problem still NP-Complete? Prove your answer.

3. The Gadget–Assembly problem stated above is the recognition version of the problem (a decision problem). An optimization version of this problem is to list the moves for a solution with minimum score. Suppose you are given a black-box, polynomial time algorithm for solving the Gadget–Assembly problem. How would you use it to get a polynomial time algorithm for this optimization version of the problem? Present the algorithm and argue that it is polynomial time.