Institut for Matematik og Datalogi
Syddansk Universitet

February 16, 2010
JFB

DM508 – Algorithms and Complexity – 2010
Lecture 6

Lecture, February 15

We continued with NP-Completeness, covering more reductions from known NP-Complete problems, showing that CLIQUE, VERTEX COVER, INDEPENDENT SET, and HAMILTONIAN CIRCUIT are NP-Complete.

Lecture, February 22

We will finish NP-Completeness and begin on amortized analysis from chapter 17 of the textbook and Fibonacci heaps from chapter 20 (chapter 19 in the third edition).

Lecture, February 24

We will finish with Fibonacci heaps (starting with the Extract-Min operation) and begin on string matching from chapter 32.

Problems to be discussed on March 2

Do problems:
17.1-2, 17.1-3, 17.2-3, 17.3-2, 17-3-7, 17-3, 3.2-7 (this is proven as Lemma 19.3 in the third edition), 20.2-1 (19.2-1 in third edition), 20.2-3, 20.2-5 (these last two are not in the third edition).

Assignment due Wednesday, March 3, 8:15

Note that this is part of your exam project, so it must be approved in order for you to take the exam in April, and you may not work with or get help from others not in your group. You may work in groups of two or three. You may write your solutions in English or Danish, but write very neatly if you do it by hand. Remember to turn it in via Blackboard.

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 evaluation 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 evaluation version of the problem? Present the algorithm and argue that it is polynomial time.