Institut for Matematik og Datalogi
Syddansk Universitet

February 13, 2008
JFB

DM508 – Algorithms and Complexity – F08
Lecture 6

Lecture, February 11

We continued with NP-Completeness, beginning on some reductions from known NP-Complete problems. We showed that Circuit Satisfiability, 3-SAT, CLIQUE, INDEPENDENT SET, VERTEX COVER, and HAMILTONIAN CIRCUIT were NP-Complete.

Lecture, February 18

We will finish with NP-Completeness, doing more reductions from known NP-Complete problems. We will also begin on amortized analysis from chapter 17.

Lecture, February 20

We will continue with amortized analysis and begin on Fibonacci heaps from chapter 20 in the textbook.

Problems to be discussed on February 26

Do problems:

1. 34.5-5 (Warning: it is tempting to think that this one is completely trivial; it is not. Also, to make this easier, you may redefine the Set Partition problem to allow the same value appearing more than once), 34.5-6.
2. 34-2, 34-3.
3. 34-1a, 34-1b, 34-1c.
4. 34-4a, 34-4b, 34-4c.
5. 17.1-2, 17.1-3

Assignment due Friday, February 29, 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 March, 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.

1. Consider the following scheduling problem: Given a set of n jobs, $J_1, J_2, ..., J_n$, where each job is given by $J_i = (r_i, d_i, t_i)$, is it possible to schedule all jobs on one machine, such that no two jobs overlap in time, and each J_i starts at or after its release time r_i, runs for time t_i, and finishes by its deadline d_i? For example, with the trivial set $\{(0, 10, 7), (2, 12, 5\}$, scheduling job J_1 to start at time 0 has it finish by time 7, so job J_2 can start and time 7 and finish by time 12. For this example, the answer would be "yes". The set $\{(0, 6, 7)\}$ cannot be scheduled. Show that this problem is NP-Complete. Hint: reduce from Subset Sum. You can add one job, not corresponding to any of the elements in the subset sum problem which can force jobs having duration adding to the target to be scheduled before the others. Most jobs will have the same release times and the same deadlines.

2. Suppose you have an algorithm A for solving the above scheduling algorithm. Give an algorithm which only makes a polynomial number of calls to A to determine a schedule that works when one exists. (It should says at which time each job starts running.)

3. Suppose G is a graph representing an adventure game, "World of Wartcraft". Each vertex in G represents a "room", and each edge represents a way of going from one room to another (i.e., a door or a magic tunnel). Each room has a leprechaun in it, who never leaves the room and is either good or bad. The first time you meet a good leprechaun, you get a gold coin. The first time you meet a bad leprechaun, you have to pay a gold coin (or you die). If you meet any leprechaun for the second time, then you die. You know the entire graph G in advance, including which rooms have good leprechauns and which have bad ones. Given an initially empty purse, the game is *winnable* if there is a way of going from a specified starting room s in G (which has a good leprechaun) to a specified ending room t without dying. Show that it is NP-Complete to determine if an instance of World of Wartcraft is winnable. Hint: reduce from Hamiltonian Path.