S-72.2420/T-79.5203 Graph Theory (5 cr) Spring 2007 Programming Project

[Figures 1 and 2 are not included in this document for copyright reasons. They are handed out during the lectures.]

Rush Hour® is a rather popular puzzle, where one tries to get a red car out of a 6×6 playing grid starting from a given position. Cars (of size 1×2) and trucks (of size 1×3) can be moved forward and backward (so the directions of the cars do not matter) to enable the red car to get out. In the sequel the word *car* is used for both cars and trucks. One example of a starting position is given in Fig. 1.

Problem 1. (Medium) Any position of the puzzle can be thought of as a vertex of a graph, with edges between positions that can be reached from one another by moving one car. Solving a problem instance is then a matter of searching this graph. Should this be done in a breadth-first or a depth-first manner? What if one is further interested in finding the smallest number of moves leading to a solution? Write a program that solves the instances in Fig. 2.

Problem 2. (Medium) Consider the graph consisting of *all* possible positions of cars with edges as discussed above. It is assumed that there are at most 4 trucks and at most 12 cars. How many such positions are there, in other words, what is the order of the graph? If you cannot obtain the exact order, can you find a lower bound on it (as large as possible, of course)?

Problem 3. (Hard) Designing puzzle instances is obviously harder than solving these. The graph theoretic framework is useful also here. How could one go about to design instances? By the solution of Problem 2, is some kind of exhaustive search of the whole graph feasible?

Note that a puzzle manufacturer has to design instances of all difficulty levels (Beginner-Intermediate-Advanced-Expert). Is the difficulty, when solving by hand, a matter of distance between the starting vertex and the final vertex; the order or size of the component in which these vertices lie; or do some other parameters matter? Are you able to find puzzle instances that are harder to solve than the hardest instance in Fig. 2 (level 40) ?

Hint 1: Some parts of the course T-79.5202 Combinatorial Algorithms might prove useful when solving these problems.

Hint 2: Problem 2 can be considered as a problem on cliques in a graph where each vertex corresponds to the position of a car and there in an edge exactly when the corresponding cars do not overlap. The software tool Cliquer (`http://users.tkk.fi/pat/cliquer.html`) for clique search may then prove useful.

The Project. The aim of this project is to solve Problems 1–3 (Problem 3 obviously does not have a concrete solution; freedom is given to what extent this problem is considered as well as to what methods are applied.) The work is done in groups consisting of 1–3 students.

The time scale for the project is as follows:

14.3. Presentation of problem, forming groups. Please register your group via e-mail to ⟨`jdubrovi@tcs.hut.fi`⟩. The e-mail should contain the names and student numbers of group members. If your group has less than 3 students, please indicate whether or not you want to be merged with another group.
16.3. Deadline for group registration.
27.4. Project review (oral presentation & written report).

Well motivated answers to the questions given earlier and a discussion of techniques used and observations made should be presented in a written report of length 5–7 pages; no source code should be appended to the report. The reports are subjected to peer review. The marking of your fellow students is in the formula for the mark of the course (see the general information of the course). The reports should be handed out during project review: one copy for each group and each of the two teachers. Each group is also to give a short oral presentation during the review describing the findings and the techniques employed.

Extended Project. It is possible to pass the whole course through a more extended treatment of this topic. The written report should then be 15-20 pages long and consist of a rigorous treatment of all three problems. The deadline for such extended reports, which will not be peer reviewed, is 15.5. A group that chooses to do the extended version of the project should tell this when registering.

Learning Objectives. Upon completion of the project, the students are able to

- model discrete problems using the language of graphs;
- apply graph software;
- collaborate with fellow students in a science project;
- report project results concisely orally and in writing;
- assess algorithms and results produced by others.