Winter term 2014/15
Professor Dr. Stephan Held
Jannik Silvanus

Research Institute for Discrete Mathematics
University of Bonn

Combinatorial Optimization

Exercise Sheet 3

Exercise 3.1:

1. Prove that a minimal factor-critical graph G has at most $\frac{3}{2}(|V(G)| - 1)$ edges and that this bound is tight. (2 Points)
2. Let G be a graph and M a matching in G. If $X \subseteq V(G)$ is the set of M-exposed vertices, then a shortest M-alternating X-X-walk of positive length can be found in $O(|E(G)|)$ (Lemma 1.38). (2 Points)

Exercise 3.2: Prove that an undirected graph G is factor-critical if and only if G is connected and $\nu(G) = \nu(G - v)$ for all $v \in V(G)$. (4 Points)

Exercise 3.4: Let $G = (V, E)$ a graph and $X \subseteq V$. Let $\beta(G, X)$ be the maximum size of a set $Y \subseteq X$ for which there is a matching in G that covers Y. Prove

$$\beta(G, X) = \min_{U \subseteq V} |X| + |U| - q_X(U).$$

Here $q_X(U)$ denotes the number of odd connected components of $G - U$ whose vertices are all in X.
Hint: Construct a new graph with $2|V|$ vertices and apply Tutte's Theorem. (4 Points)

Deadline: Tuesday, October 28, 2014, before the lecture.
Information: submissions by groups of one or two students are allowed.
Note the programming exercise on page 2!

Programming Exercise 1

Implement Edmond's Cardinality Matching Algorithm.

Program Specification: Your program must accept a filename as a command-line parameter (i.e. it must be called with `myprogram input.dmx`). The command-line parameter contains the filename of the file that encodes the graph.

Input: The input file is a DIMACs file that encodes an undirected graph. That means, the first line has the format

`p edge` n m

where n is the number of vertices of the graph and m is the number of edges. The following m lines have the format

`e` i j

where i and j are the indices of the vertices connected by this edge. The vertices are indexed from 1 to n. Lines starting with a `c` are comments and should be ignored. For a more complete definition of the DIMACS format, see `http://www.or.uni-bonn.de/lectures/ss12/praktikum/ccformat.pdf`. For testing purposes, you can use the files at `http://www.or.uni-bonn.de/lectures/ss12/praktikum/index.html`. You may use code fragments of the DIMACS reader from `mss.c` in `http://www.or.uni-bonn.de/~held/lpip/1314/mss.zip`.

Output: Your program must write the matching, encoded in the DIMACS format, to the standard output.

Programming Languages: Your program must be written in C or C++ and compile with a GNU compiler on a current Linux machine.

Criteria: The following criteria are relevant for the number of points you will be awarded: Correctness, speed, code documentation, number of compiler warnings, overall elegance.

(20 Points)

Deadline: Thursday, November 14, 2014, before the lecture. Send your program to `silvanus@or.uni-bonn.de`.