Winter term 2012/13
Juniorprofessor Dr. Stephan Held
Jan Schneider

Research Institute for Discrete Mathematics
University of Bonn

Combinatorial Optimization

Exercise Sheet 4

Exercise 4.1:
Prove: A graph G has a perfect matching if and only if for each $X \subseteq V(G)$, the graph $G - X$ has at most $|X|$ factor-critical components.
(1 Point, 2 extra points if you do not use the Gallai-Edmonds Theorem)

Exercise 4.2:
Let $G = (V, E)$ with $|V| = 2k$ and $|\delta(v)| \geq k$ for all $v \in V$. Show that G has a perfect matching.
(3 Points)

Exercise 4.3:
Let G be a k-connected graph with $2\nu(G) < |V(G)| - 1$.

1. Prove $\nu(G) \geq k$. (2 Points)
2. Prove $\tau(G) \leq 2\nu(G) - k$. (2 Points)

Exercise 4.4:
Let G be a bipartite graph with $n := |V(G)|$ and $m := |E(G)|$.

1. Prove that, given a matching M in G, the union of all shortest M-augmenting paths in G can be found in $\mathcal{O}(m + n)$ time.
   Hint: Use a variant of breadth-first search. (2 Points)
2. Consider a sequence of iterations of the algorithm where the length of the augmenting path remains constant. Show that the time needed for the whole sequence is no more than $\mathcal{O}(m + n)$.
   Hint: Use (1.) and apply a variant of depth-first search. (2 Points)
3. Describe an algorithm with runtime $\mathcal{O}(\sqrt{n}(m + n))$ that solves the Cardinality Matching Problem in bipartite graphs. (2 Points)

Deadline: Tuesday, November 6, 2012, before the lecture.

Note the programming exercise on page 2!

Programming Exercise 1:

Implement Edmonds' 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, one 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. After this line, 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 base your file parser on the code available at `http://www.or.uni-bonn.de/lectures/ws11/lgo_uebung_ws11.html`.

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.

Submission: Send your program to `schneid@or.uni-bonn.de`.

(20 Points)

Deadline: Tuesday, November 27, 2012, before the lecture.