Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institut für Theoretische Informatik
Dr. Tibor Szabó and Miloš Stojaković

November 17, 2004

Graph Theory — Problem Set 5

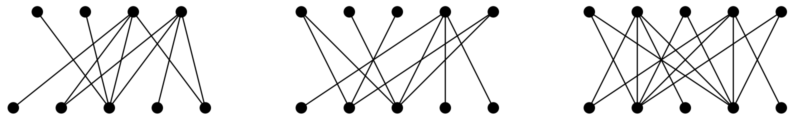
Course Webpage: `http://www.ti.inf.ethz.ch/ew/courses/GT04/`

Due Date: November 24, 2004 at the lecture

Exercise 5.1

(Exercise 3.1.1 in the Textbook)

(−) Find a maximum matching in each graph below. Prove that it is a maximum matching by exhibiting an optimal solution to the dual problem (minimum vertex cover). Explain why this proves that the matching is optimal.



Exercise 5.2

(Exercise 3.1.7 in the Textbook)

(−) Prove that a graph G is bipartite if and only if $\alpha(H) = \beta'(H)$ for every subgraph H of G with no isolated vertices.

Exercise 5.3

(Exercise 3.1.8 in the Textbook)

(!) Prove or disprove: Every tree has at most one perfect matching.

Exercise 5.4

(Exercise 3.1.9 in the Textbook)

(!) Prove that every **maximal** matching in a graph G has at least $\alpha'(G)/2$ edges.

Exercise 5.5

(Exercise 3.1.24 in the Textbook)

(!) A **permutation matrix** P is a $0,1$-matrix having exactly one 1 in each row and column. Prove that a square matrix of nonnegative integers can be expressed as the sum of k permutation matrices if and only if all row sums and column sums equal k.

Exercise 5.6

(Exercise 3.1.32 in the Textbook)

(!) In an X,Y-bigraph G (namely, a bipartite graph with X and Y as its partite sets), the **deficiency** of a set S is $\mathrm{def}(S) = |S| - |N(S)|$; note that $\mathrm{def}(\emptyset) = 0$. Prove that

$$\alpha'(G) = |X| - \max_{S \subseteq X} \mathrm{def}(S).$$

(Hint: Form a bipartite graph G' such that G' has a matching that saturates X if and only if G has a matching of the desired size, and prove that G' satisfies Hall's Condition.) (Ore [1955])

Exercise 5.7

(Exercise 2.3.14 in the Textbook)

(!) Let C be a cycle in a connected weighted graph. Let e be an edge of maximum weight on C. Prove that there is a minimum-weight spanning tree not containing e. Use this to prove that iteratively deleting a heaviest non-cut-edge until the remaining graph is acyclic produces a minimum-weight spanning tree.