Diskrete Mathematik II WS 2012 / 2013
Tibor Szabó

Exercise Sheet 2

Due date: Oct 30, 12:30 PM, beginning of exercises
NO LATE SUBMISSIONS!

You should try to solve and write up all the exercises. You are welcome to submit **at most** two neatly written exercises each week. You are encouraged to submit in pairs, but don't forget to mark the name of the scriber.

Exercise 1.
Show that for any bipartite graph $G = (X \cup Y, E)$ there exists a subset $S \subseteq X$ such that $\alpha'(G) = |X| - |S| + |N(S)|$.

Exercise 2.
Let $G = (X \cup Y, E)$ be a bipartite graph with $|X| = m, |Y| = n$. Assume $|N(S)| \geq |S|$ for any $S \subseteq X$. Set $\delta := \min\{\deg(x) : x \in X\}$. Show that G contains at least

$$\prod_{0 \leq i < \min\{\delta, m\}} (\delta - i) \tag{1}$$

distinct matchings saturating X.

Exercise 3.
An island of area n has n married hunter/farmer couples. The Ministry of Hunting divides the island into n equal-sized path connected [1] hunting regions. The Ministry of Agriculture divides it into n equal-sized path connected farming regions. However, the Ministry of Marriage requires that each couple receives two overlapping regions. Prove that it is always possible to assign to each couple one hunting region and one farming region in such a way that they overlap.

Bonus question: Prove the following stronger result: there is always an assignment where each couple's two regions share an area of size at least $4/(n+1)^2$, if n is odd, or at least $4/[n(n+2)]$, if n is even. Furthermore, prove that the lower bound on the size of the intersections is best possible.

Exercise 4.
A company decides to reward its n employees for working overtime. Every day, for every employee it records the hour he came at the office (which we shall call *entry time*), and the

[1] there is a continuous curve connecting any two points of the region.

hour he left the office (which we shall call *exit time*). Every employee is paid an additional bonus of x euros every hour he/she spends over his/her usual 8 hours.

Unfortunately, at the end of one day the computer keeping the entry and exit times broke, and instead of outputting n pairs of the form *(entry time, exit time)*, one for each employee, it output two scrambled arrays, each of n numbers, the first representing the entry times in no particular order, and the second representing the exit times in no particular order.

The company would now like to reconstitute the n pairs *(entry time, exit time)*, subject to the obvious restriction *exit time* $>$ *entry time*. The company is obviously interested in minimizing the total amount of money paid to its employees for their overtime. It is allowed that an employee spends less than 8 hours at the office (though in this case he will certainly receive no overtime bonus).

Please describe an algorithm that resolves this problem. Argue the correctness of the solution.