Exercise sheet 2

Exercises for the exercise session on 12/04/2018

Problem 2.1. Let G be a 2-connected graph and let $e \in E(G)$.

(a) Prove that all ear-decompositions of G have the same number k of ears.

(b) Show that there are ear-decompositions $C, P_1, \dots, P_k$ and $\tilde{C}, \tilde{P}_1, \dots, \tilde{P}_k$ of G such that e lies on C and on $\tilde{P}_1$.

(c) Prove that e lies on at least $k+1$ distinct cycles in G.

Problem 2.2. Prove that the following statements are equivalent for every non-empty graph G.

(i) No two cycles of G intersect in more than one vertex;

(ii) no edge of G lies on more than one cycle;

(iii) each block of G is either an isolated vertex, a bridge, or a cycle.

Note. A connected graph satisfying these properties is called *cactus*.

Problem 2.3. Prove that every graph G with at least two vertices satisfies

$$\kappa(G) \leq \lambda(G) \leq \delta(G).$$

For every integer $k \geq 1$, construct graphs G_1, G_2 with

$$\kappa(G_1) = 1, \quad \lambda(G_1) = \delta(G_1) = k,$$
$$\kappa(G_2) = \lambda(G_2) = 1, \quad \delta(G_2) = k.$$

Bonus problem. If d, k, l are integers with $1 \leq k \leq l \leq d$, is there always a graph G with $\kappa(G) = k$, $\lambda(G) = l$, and $\delta(G) = d$?

Problem 2.4. For a graph G, its *line graph* $L(G)$ is defined as the graph on vertex set $E(G)$, in which distinct $e, e' \in E(G)$ are adjacent (as vertices of $L(G)$) if and only if they intersect (as edges of G).
Use $L(G)$ to prove the edge version of Menger's theorem: For disjoint sets A, B of vertices in a graph G, the largest number of edge-disjoint A–B paths equals the smallest size of an edge set separating A and B.

Problem 2.5. Let G be a bipartite graph with sides A and B.

(a) Let M_A, M_B be matchings in G. Denote by A' the set of vertices in A that M_A covers; define B' analogously for M_B and B. Prove that G has a matching that covers $A' \cup B'$.

(b) Use (a) to show that G has a matching that covers all vertices of maximum degree $\Delta(G)$. Deduce that every r-regular bipartite graph (with $r \geq 1$) has a perfect matching.

Problem 2.6. For a bipartite graph G, consider the algorithm from the lecture that constructs a largest matching in G by recursively finding augmenting paths via BFSm.

(a) Prove that if M is not largest possible, then BFSm finds an unmatched vertex in B.

(b) Suppose (for simplicity) that $|A| = |B|$ and determine (the order of) the running time depending on $n := |G|$ and $m := \|G\|$. What is the running time if we know that a largest matching consists of μ edges? Simplify the formulas under the additional assumption that $m = \Omega(n)$.