# Advanced and algorithmic graph theory

Summer term 205

## Exercise sheet 2

Exercises for the exercise session on 07/04/2025

(Bonus problems are not counted towards the total number of problems, but solving a bonus problem will earn you a bonus point.)

**Problem 2.1.** Let $G$ be a 2-connected graph which is not a cycle 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 $C, P_1, \ldots, P_k$ and $\tilde{C}, \tilde{P}_1, \ldots, \tilde{P}_k$ 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 $G$.

**Bonus problem.** Is the statement of Problem 2.1(b) best possible? In other words, does there exist, for every choice of integers $k \geq j \geq 2$, a 2-connected graph $G$ and an edge $e \in E(G)$ such that every ear-decomposition of $G$ is of the form $C, P_1, \ldots, P_k$, but no such ear-decomposition satisfies $e \in E(P_j)$?

**Problem 2.2.** Design an algorithm that constructs ear-decompositions of 2-connected graphs. What running time can you achieve?
*Note. Do not write (pseudo-)code for your algorithm, but rather describe in words which steps should be used to find the cycle and the ears of the ear-decomposition.*

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

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

Furthermore, for all integers $d, k, l$ with $1 \leq k \leq l \leq d$, find 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) in $L(G)$ if and only if they are adjacent (as edges) in $G$.
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 indeed finds an unmatched vertex in $B$ (and thus an augmenting path).

(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 $\mu$ edges? Simplify the formulas under the additional assumption that $m = \Omega(n)$.