Prof. Dr. Sándor P. Fekete
Peter Kramer

Due May 13^{th}
Discussion May 20^{th}

Please submit your handwritten answers in pairs*, using the box in front of* `IZ338` before *the exercise timeslot on the due date above. Make sure to include your* full names*,* matriculation numbers*, and the* study programmes *that you are enrolled in (e.g., computer science master).*

Exercise 1 (Minimum-Cardinality Maximal Matching). **(12 + 3 points)**

In the lecture, we used a maximal matching to approximate the Vertex Cover problem. Maximality of the matching is necessary to guarantee feasibility of the resulting vertex cover. This motivates the search for the smallest maximal matching of a given undirected graph G.

a) Show that, for any two maximal matchings $A, B \subseteq E(G)$ with $|A| \leq |B|$, $|B| \leq 2|A|$.
(Hint: Assume that $|A| < {|B|}/{2}$ and consider the number b of edges in B that are incident to a vertex that is matched in A.)

b) Give a 2-approximation algorithm for the problem of finding a smallest maximal matching.

Exercise 2 (Bin Covering). **(7 + 4 + 4 points)**

In the big tutorial, we saw the Bin Packing problem, where we are given a set of rationals $z_1, \ldots, z_n$ with $0 < z_i < 1 \in \mathbb{Q}$ and are asked to pack these into the smallest possible number k of bins $B_1, \ldots, B_k$ with $B_i \subset \{1, \ldots, n\}$ such that each bin has sum at most 1, i.e.,

$$\forall i = 1 \ldots n : \quad \sum_{j \in B_i} z_j \quad \leq 1.$$

In this exercise, we consider the opposite problem, Bin Covering: we want to maximize the number of bins that contain items of sum at least 1.

a) Formulate an integer linear program (ILP) for Bin Covering. Explain the number and role of your variables and constraints, and give a *brief* argument for correctness.

The following greedy algorithm for Bin Covering considers the items in their given order and opens a new bin as soon as the current one is full.

b) Prove that Algorithm 1 is *at best* a 2-approximation for arbitrary input size n.

c) Prove that Algorithm 1 is a 2-approximation.

Algorithm 1: Greedy Bin Packing

```
B_1 ← an empty bin
k ← 1
for z_i in z_1, ..., z_n do
    B_k ← B_k ∪ {i}
    if ∑_{j∈B_k} z_j > 1 then
        k ← k + 1
        B_k ← an empty bin
    end
end
output B_1, ..., B_k
```