University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
P. Bamberger, P. Schneider

Algorithm Theory
Exercise Sheet 7

Due: Tuesday, 7th of December, 2021, 4 pm

Exercise 1: Vertex Cover Variant *(10 Points)*

Given an undirected graph $G = (V, E)$, a subset $U \subseteq V$ of nodes and a capacity function $c : U \to \mathbb{N}$, we want to cover every edge with the nodes in U, where every node $u \in U$ can cover up to $c(u)$ of its incident edges.
Formally, we are interested in the existence of an assignment $f : E \to U$ such that for all $e \in E$ we have $f(e) \in e$ and for all $u \in U$ it holds $|\{e \in E \mid f(e) = u\}| \leq c(u)$.

Devise an efficient algorithm to determine whether or not such an assignment exists and explain its runtime.

Exercise 2: Cycle Elimination *(10 Points)*

Let $G = (V, E, c)$ be a directed graph with capacity function $c : E \to \mathbb{N}$ and let $s, t \in V$. We allow G to contain cycles. We now want to build a DAG (directed acyclic graph) $G' = (V, E', c')$ with $E' \subseteq E$ and $c'(e) = c(e)$ for $e \in E'$ (i.e., we obtain G' by deleting edges from G) that has the same minimum s-t cut capacity as G.

Give an efficient algorithm to compute such a graph G', argue that your algorithm is correct and analyze its runtime.