

# Exercise Sheet 7 for Advanced Data Structures (Summer 2026)

**Hand In:** Until 2026-06-12 18:00, on ILIAS.

## Problem 1

60 points

You are given a connected graph  $G = (V, E)$ , with vertex weights  $w : V \rightarrow \mathbb{N}$  and edge costs  $c : E \rightarrow \mathbb{N}$ .  $G$  has  $n$  vertices and  $m$  edges. For any value  $c^* \in \mathbb{N}$ , we define  $G[c^*]$  to be the subgraph of  $G$  containing only edges  $(u, v) \in E$  with  $c(u, v) \leq c^*$ .

You are given  $q$  queries  $(v_i, k_i)$  for  $i = 1, \dots, q$ . For each query  $(v_i, k_i)$ , output  $c_i^*$ , the smallest integer such that the connected component of  $v_i$  within  $G[c_i^*]$  has at least  $k_i$  total weight. (The total weight of a connected component is the sum of the weights of all the vertices within that connected component.)

For full marks, your algorithm should run in  $O(m \log m + q \log q)$  time. Note that the queries are given *offline*, i.e. all are given as a batch.

*Hint:* Sort the edges from small weight to large weight, and add the edges in as you go along. Try to answer queries for every connected component every time these change.