Exercise set 7: Review questions

Question 1. True or false? For each statement below decide whether it is true for all positive functions $f(n)$, $g(n)$ and $h(n)$, or give a counter example.

1. If $f - g = O(1)$, then $\frac{f}{g} = O(1)$.
2. $f + g = \Theta(\max\{f, g\})$.
3. $f + g = \Theta(\min\{f, g\})$.
4. For any constants $a, b > 0$, $(n + a)^b = \Theta(n^b)$.
5. If $f = O(g)$ and $g = O(h)$ then $f = O(h)$.
6. If $f = O(g)$, then $f + g = \Theta(g)$.
7. If $f = O(g)$, then $f \cdot g = \Omega(f)$.

Question 2. For each of the following problems:

- Give a greedy algorithm that finds an optimal solution.
- Prove that the output of your algorithm is valid.
- Prove that the output of your algorithm is optimal.
- Give a nontrivial example and show that your algorithm indeed outputs the optimal solution on that example.

The Cupcake Problem: Given are n cupcakes of different weights and different qualities. For a party, we want to put several cupcakes on a single tray such that the overall quality of the cupcakes on the tray is maximal, while the total weight of the tray does not exceed a certain limit. Formally, the i-th cupcake has weight $w_i \in \mathbb{N}$ and quality $q_i \in \mathbb{N}$, and there is a total weight limit of $G \in \mathbb{N}$. We are allowed to cut the cupcakes, so the problem is to find fractions $\alpha_1, \ldots, \alpha_n \in [0, 1]$, with $\alpha_1 w_1 + \cdots + \alpha_n w_n \leq G$, such that the overall quality $\alpha_1 q1 + \cdots + \alpha_n qn$ is as large as possible.

Matching Skiers and Skis: A skier of height h with a pair of skis of length l has a discrepancy of $|h - l|$. *Input:* a set of n skiers of heights $h_1, h_2, \ldots, h_n$, and a set of n skis of length $l_1, l_2, \ldots, l_n$. *Output:* A matching between the

n pairs of skis and the n skiers that minimizes the maximum discrepancy.

Question 3. Given is an array $A = (a_1, a_2, \ldots, a_n)$ of n integers. We define $count(x, A)$ as the number of occurrence of x in A. The majority element of the array is some x such that $count(x, A) > n/2$ (if such an element exists) or NULL otherwise. The problem is to find the majority element in A.

E.g., the majority elements of the arrays, of size 10, $A_1 = (3, 2, 11, 11, 11, 2, 15, 11, 11, 11)$ and $A_2 = (3, 3, 3, 3, 8, 8, 4, 4, 4, 4)$ are 11 and NULL, respectively.

- Devise an algorithm that finds the majority element with running time $O(n^2)$. Verify correctness and running time.
- Devise an algorithm that finds the majority element with running time $O(n \log n)$. Verify correctness and running time.