Introduction to Computer Science
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: CH-232-A
Date: 2020-09-18
Due: 2020-09-25

ICS 2020 Problem Sheet #2

Problem 2.1: *time complexity and landau sets* (1+1+2 = 4 points)

A program consists of two parts that are executed sequentially. The size of the input is described by n. The first part has a time complexity that can be expressed as $t_1(n) = 5n^2 + 16$ and the second part has the time complexity $t_2(n) = 6n^3 + n^2 + 18$. Use only the definition of the big O notation, do not use any laws for the big O notation (unless you prove them as well).

a) To which big O sets do t_1 and t_2 belong?

b) To which big O set does the entire program belong?

c) Prove that if $f_1 \in O(g_1)$ and $f_2 \in O(g_2)$, then $(f_1 + f_2) \in O(\max\{g_1, g_2\})$.

Problem 2.2: *proof by induction* (4 points)

Let n be a natural number with $n \geq 1$. Prove that the following holds:

$$1^2 + 3^2 + 5^2 + \ldots (2n-1)^2 = \sum_{k=1}^{n}(2k-1)^2 = \frac{2n(2n-1)(2n+1)}{6}$$

Problem 2.3: *list comprehensions (haskell)* (1+1 = 2 points)

Your list comprehensions should be correct, they do not have to be efficient. You are not getting points for a list comprehension simply returning a hard coded solution list. In other words, your list comprehensions should continue to function correctly if parameters are changed.

a) Write a list comprehension that returns all positive factors of the number 210. Try to write the list comprehension in such a way that 210 can easily be replaced by a different number.

b) Write a list comprehension that returns a list of Pythagorean triads (a, b, c), where a, b, c are positive integers in the range $1..100$ and the Pythagorean triad is defined as $a^2 + b^2 = c^2$. The list should not contain any “duplicates” where a and b are swapped. If the list contains $(3, 4, 5)$ (since $3^2 + 4^2 = 25 = 5^2$), then is should not also include $(4, 3, 5)$.