CS340 – Advanced Data Structures and Algorithm Design – Fall 2020
Handout 3 – October 09, 2020

Dr. Sandra Zilles, Department of Computer Science, University of Regina

preparation for midterm exam

Note. The midterm exam will take place on Friday, **October 16**, in the usual class zoom room, starting at **11.30am**. The duration of the exam is **50 minutes**. This exam is **closed-book** – no auxiliary material is allowed. Attempted cheating in the midterm exam may lead to a fail grade for the whole course. Note that the exam will be proctored using ProctorTrack.

Material covered. The material about which exam questions may be asked includes all the material that was covered in lectures or assignments up to and including October 14. For example, this contains material including (but not limited to) the following.

> **Chapter 1. Algorithm Analysis.** Which model do we use to analyze algorithms? How is the running time of an algorithm defined? What is worst case running time, what is average case running time, what is the best case running time? What is amortized running time and how is it calculated? How can we compare two algorithms in terms of running time efficiency? How do different classes of growth rates of functions compare asymptotically? What do the asymptotic notations O, o, Ω, and Θ mean and how are they defined? What are their most important properties? How do we determine which O, o, Ω, Θ classes a given function belongs to? How do we analyze a simple program fragment and determine its running time complexity? How do we determine how often the body of a loop is executed? How does a proof by induction work?
>
> **Excursion. Splay Trees.** What is a splay tree, how do we implement it? What are the most efficient algorithms for basic operations on splay trees and how did we analyze their running time cost (and with which result)? Illustrate all operations on example trees! How do Splay Trees relate to AVL Trees?
>
> **Chapter 2. Priority Queues.** What is a priority queue and for what type of application is it needed? What is a binary heap and how do we implement it? What are the most efficient algorithms for basic operations on binary heaps and how did we analyze their running time cost (and with which result)? Illustrate all operations on example binary heaps! How would you answer the previous questions for d-heaps, leftist heaps, skew heaps, binomial queues (instead of binary heaps)?
>
> **Chapter 3. Sorting.** How does insertion sort work? How did we analyze it (and with which result)? How to illustrate insertion sort for a small example? How does Shellsort work? How to illustrate Shellsort for a small example?

Typical questions. Questions may, for example, be of the following types (not exclusively).

- questions similar to those in the theory parts of the assignments or to the theory questions in the textbook chapters we covered in class,
- questions asking to explain or define concepts or results introduced in class,
- multiple-choice questions,
- questions asking for algorithms in pseudocode, related to tasks discussed in class or in the assignments,
- questions about which data structure or which algorithm to prefer in a particular situation,
- proofs of formal statements,
- ...

The questions will be of varying difficulty. For every question it will be possible to achieve partial marks. Even when you have only a partial idea, it is advisable to type it into UR Courses.