CS340 – Advanced Data Structures and Algorithm Design – Fall 2020
Assignment 3 – September 25, 2020

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

due October 5, 2020, 10.00 am

- Please submit one single pdf for your solution to all problems, including code printouts and printouts of results from running your code, through URCourses.
- Please submit your `C++` code (compilable in Visual Studio or g++) for the solution of the programming part in UR Courses, in addition to the printed version of the code. In UR Courses, please submit everything related to your code in a single zip file.

Problem 1 (3+3+1 marks).

(a) Show the result of inserting 10, 12, 1, 14, 6, 5, 8, 15, 3, 9, 7, 4, 11, 13, and 2, one at a time, into an initially empty binary heap. Additionally, show the intermediate result after every single insertion. (This is an extension of textbook problem 6.2(a).)

(b) Show the result of using the linear-time algorithm to build a binary heap using the same input. (This is textbook problem 6.2(b).)

(c) Show the results of two consecutive `deleteMin` operations on the 3-heap in Figure 6.19 in the textbook.

Problem 2 (2+3 marks).

(a) Show that, in any heap, the maximum item must be at one of the leaves. (This is textbook problem 6.8(a).)

(b) Show that any binary heap contains exactly $\lceil N/2 \rceil$ leaves. (This is textbook problem 6.8(b).)

Problem 3 (4+2+2 marks).

(a) Implement the following algorithm, which is given a duplicate-free array `array` as input, in `C++`.
`whatDoIDo (array):`

1) Build a heap from `array` (using `buildHeap` as explained in class), where the heap starts at position `array[0]`.
2) Starting from `j` = size of `array - 1`, as long as `j>0`:
   i. Swap the entries `array[0]` and `array[j]`.
   ii. Percolate down `array[0]`, but only within the subarray `array[0..j-1]`.
   iii. Decrement `j` by `1`.

Provide three input/output examples for duplicate-free arrays of size about 10.

(b) What does `whatDoIDo` do? Explain your answer.

(c) What is the worst-case running time of `whatDoIDo`, in dependence of the size N of the given array? Explain your answer.