Assignment 3

Grades: each assignment is 5% in the final score and there is a 1% bonus in each assignment for bonus problems. We use the following scale for the simplicity of grading: the total score is 50 (additional 10 for bonus problems) for each assignment.

Problem 1 [Full Score: 10]. Show the contents of an array $A$ implementing a min-heap. Repeatedly insert the following values into an initially empty heap: 7, 6, 5, 4, 3, 2, 1. Use the up-heap bubbling insertion of the items one-by-one. (Show the content after each insertion.)

Problem 2 [Full Score: 10]. Show the contents of the array $A$ as a binary min-heap, initially with entries [7, 6, 5, 4, 3, 2, 1], as it changes during the linear time bottom-up heap construction (i.e., Build-Min-Heap()). (Show the content after each down-heap bubbling, i.e., Min-Heapify().)

Problem 3 [Full Score: 10]. Let $T$ be a min-heap storing $n$ keys. Give an efficient algorithm for reporting all the keys in $T$ that are no more than a given query key $k$ (which is not necessarily in $T$). Ideally, your algorithm should run in $O(m)$ time, where $m$ is the number of keys reported. (You need to describe your algorithm and show why its complexity is $O(m)$.)

Problem 4 [Full Score: 10]. Given $k$ sorted queues (i.e., the front of each queue stores the minimum element in each queue) containing a total of $n$ elements, describe how to merge them into a single sorted queue in time $O(n \log k)$. You need to describe your algorithm and prove its complexity. (Hint: use a min-heap.)

Problem 5 [Full Score: 5]. Provide a brief explanation for Exercise 6.3-2 (on page 159).

Problem 6 [Full Score: 5]. Exercise 6.5-8 (on page 166).

Problem 7 [Bonus Problem: 10].

1) Develop an algorithm that computes the $k$th smallest element of a set of $n$ distinct integers in $O(n + k \log n)$. (You need to describe your algorithm and show why its complexity is $O(n + k \log(n))$.)

2) Describe a data structure that supports both Heap-Extract-Min() and Heap-Extract-Max() with $O(\log(n))$ complexity. You need to show the implementation of Heap-Insert(), Heap-Extract-Min(), and Heap-Extract-Max(), and prove their complexity. Hint: one can build on top of the heap.