CMPSC 465 Fall 2025 — Data Structures & Algorithms — Ke Chen and Yana Safonova

HW 3

Due September 29, 10:00 pm

Instructions: You are encouraged to solve the problem sets on your own, or in groups of three to five people, but you must write your solutions strictly by yourself. You must explicitly acknowledge in your write-up all your collaborators, as well as any books, papers, web pages, etc. you got ideas from.

Formatting: Each part of each problem should begin on a new page. Each page should be clearly labeled with the problem number and the problem part. The pages of your homework submissions must be in order. When submitting in Gradescope, make sure that you assign pages to problems from the rubric. You risk receiving no credit for it if you do not adhere to these guidelines.

Late homework will not be accepted. Please, do not ask for extensions since we will provide solutions shortly after the due date. Remember that we will drop your lowest three scores.

This homework is due Monday, September 22, at 10:00 pm electronically. You need to submit it via Gradescope. Please ask on Canvas about any details concerning Gradescope.

1. (20 pts.) **Heap Basics.** The array $A := [23, 4, 14, 3, 9, 13, 11, 10]$.

(a) Run build-heap on A to construct a min-heap and write down the resulting array.

(b) Insert the element 5 and write the resulting array.

(c) Insert the element 1 and write the resulting array.

(d) Delete 14 and write the resulting array.

2. (30 pts.) **Heaps and Heap Sort.**

(a) What are the minimum and maximum numbers of nodes in a heap of height h?

(b) Is the array with values $\{10, 14, 19, 35, 31, 42, 27, 44, 26, 33\}$ a Min heap?

(c) Show that in the worst-case Heapify-UP could make $\Omega(\log n)$ swaps on a heap with n elements. (Hint: Give an example heap with n node values that would cause Heapify-UP to be called recursively at every node on a simple path up to the root).

3. (30 pts.) **Median of Streaming Data.** Consider an array of N numbers $[x_1, x_2, \ldots, x_N]$ that you will be receiving one-by-one in a single pass (i.e., as a stream of numbers). You are not allowed to revisit previous numbers. Design an effective data structure involving one or more heaps that, after receiving the n-th number, reports the median of the numbers $x_1, x_2, \ldots, x_N$ observed so far. The time complexity of your algorithm should be $O(\log n)$ per number received in the worst case. Write down the pseudocode of your algorithm and analyze its time complexity. For simplicity, assume there are no duplicates in the stream. Hint: If n is odd, the median of the data stream $[x_1, x_2, \ldots, x_N]$ is the middle element of the sorted data stream, else the median is the average of the middle two elements of the sorted data stream.

4. (20 pts.) **Proofs.** Show that an n-element binary heap has height $\lfloor \log_2 n \rfloor$.