ELEC 278: Fundamentals of Information Structures
Lab 6: Priority Queues and Heaps

Fall 2023–Instructors: Ni & Mertin

November 17, 2023

Please read the entire document to understand the requirements and process for completing the lab.

1 Objectives

The objectives for this lab are for you to demonstrate your understanding of priority queues as a concept and the implementation of priority queues as heaps in C. Starting from a provided implementation of a binary heap data structure, you will make changes to the implementation as specified below.

2 Instructions

Download the file `lab6.zip` from OnQ and unzip it. Open the `lab6` folder that you extracted in either CLion or VS Code (note: you need to make sure you open the correct folder, which is, the one that directly contains the code files). Then, complete the following tasks.

This lab is due 11:30 AM (end of the lab session) on Thursday, November 30 for students in all sections. The scheduling of lab sessions is as follows:

- Tuesdays 12:30–2:30
- Wednesdays 11:30–1:30
- Thursdays 9:30–11:30

The provided code implements a binary max-heap data structure, complete with insertion and removal algorithms. You are asked to make the following two changes to the provided code:

- It should be changed to a ternary heap instead of a binary heap. This means that each node has (up to) 3 children instead of 2. None of the other heap properties change; it is still the case that each node should have a higher priority than all of its children. You must translate what you have learned about binary heaps, and what you see already implemented in front of you, to the idea of a ternary heap.
- It should be changed to implement a min-heap instead of a max-heap; i.e., it should consider elements with *lower* priority numbers to be higher priority.

This is designed as a simulation of a real-world task that a software engineer may be asked to do, within the constraints of what you have been taught in the course. As a result, it is important that you stick to the overall structure of the provided code; creating a new implementation from scratch will not be acceptable.

You should approach this lab by first reading through the existing code to understand how it works, then planning the changes (such as figuring out on paper how to compute indices for a ternary heap), then implementing the changes in code. Finally, test the resulting code to make sure the modified data structure works as expected.

3 Marking Criteria

After completing all tasks, call over a graduate TA to mark the lab. Lab 6 has 5 marks in total:

- Does your modified data structure correctly implement a ternary heap? Are all elements in the correct place? (3 marks)
- Does your modified data structure correctly order elements in a min-heap? (1 marks)
- Is your code sufficiently well-formatted and commented so that the purpose of each variable/field/parameter and the reason for each function call or data structure manipulation is clear? Refer to the guidance on the course assignment, as well as the starter code for this lab, for expected levels of commenting. (1 marks)