Assignment 2 — Heap Data Structures

Comparison Summary: MinHeap vs MaxHeap

Authors: Ernar Sadenov & Yerkebulan Sovet | SE-2430

Date: 05.10.25

1. Introduction

This document provides a joint comparison of MinHeap and MaxHeap implementations. Both are binary heap data structures that support efficient priority queue operations. They are symmetric in design but differ in whether the smallest or largest element is stored at the root.

2. Theoretical Comparison

- MinHeap: The root is the minimum element; useful when smallest values must be accessed quickly.

- MaxHeap: The root is the maximum element; useful when largest values are needed.

Complexities (identical):

- Insert: O(log n)

- Extract (Min/Max): O(log n)

- Build Heap: Θ(n)

- Space: O(n)

Difference: only the comparison operator (`<` vs `>`).

3. Empirical Performance

Benchmark tests with inputs ranging from 100 to 100,000 elements confirm that both MinHeap and MaxHeap perform almost identically. Log-log runtime plots align with theoretical expectations. Minor runtime variations occur due to input randomness but remain negligible.

4. Applications

- MinHeap: Dijkstra’s shortest path, Prim’s MST, priority queues for smallest values.

- MaxHeap: Heap Sort, task scheduling with highest-priority tasks.

5. Conclusion

Both MinHeap and MaxHeap show the same efficiency. Their choice depends on the problem: whether quick access to the minimum or maximum element is required. Empirical and theoretical analyses confirm their symmetry.