

# DAA Assignment 2 — Max-Heap Peer Review Report

## 1. Overview

This report reviews the Max-Heap implementation submitted by the partner for Assignment 2. The project includes a MaxHeap class, a PerformanceTracker for counting operations, a BenchmarkRunner for measuring performance, and unit tests in MaxHeapTest.

## 2. Implementation Summary

The Max-Heap uses an array-based structure with standard operations: insert, extractMax, peek, and increaseKey. The code follows the Floyd bottom-up build for constructing a heap efficiently in  $\Theta(n)$  time. PerformanceTracker correctly counts swaps, comparisons, and array accesses, which helps in empirical evaluation.

## 3. Theoretical Complexity

Operation	Time	Space	Notes
insert	$O(\log n)$	$O(n)$	Move up element
extractMax	$O(\log n)$	$O(n)$	Move down element
peek	$O(1)$	$O(1)$	Return top element
build (Floyd)	$\Theta(n)$	$O(n)$	Bottom-up heap construction
increaseKey	$O(\log n)$	$O(1)$	Adjust element upward

## 4. Strengths

- The implementation is clean and well-structured.
- The code correctly separates algorithm logic from metric tracking.
- BenchmarkRunner automates CSV export, which is helpful for analysis.
- Unit tests cover main operations and check edge cases such as extracting from an empty heap.

## 5. Suggestions for Improvement

These suggestions focus on improving code clarity and small readability details rather than algorithmic changes.

1. **Clean Code:** Add short comments above each main method (insert, extractMax, peek) describing what it does.
2. **Consistent Naming:** Use clear and consistent variable names such as ``heap``, ``size``, and ``capacity`` throughout the code.

3. **Formatting:** Keep the same indentation (4 spaces) and spacing between methods for readability.
4. **Avoid Repetition:** Combine similar print or test statements into helper functions where possible.
5. **Error Handling:** Add clear exception messages when the heap is empty (e.g., "Cannot extract from an empty heap").
6. **Metrics Clarity:** Print metrics in a readable format (for example: "Comparisons: 1250, Swaps: 300").
7. **README Update:** Include instructions on how to run BenchmarkRunner and where the CSV file is generated.

## **6. Overall Evaluation**

The Max-Heap implementation meets assignment requirements and demonstrates good programming practice. The structure is easy to follow, and performance metrics make it suitable for empirical validation. Minor improvements in code clarity and consistency will make the project more professional and easier to maintain.