**Lab 3**

**Implementing Binary Heap &**

**Sorting Techniques**

|  |  |
| --- | --- |
| Name | **ID** |
| Muhammad El Kotb | 19016258 |
| Ahmed Adel Abudef | 19015264 |

**Table Of Contents**

[1. Problem Statement 2](#_Toc98185149)

[2. UML Diagram 3](#_Toc98185150)

[3. Analysis 3](#_Toc98185151)

# Problem Statement

* **Max-Heap**

In this assignment, you’re required to implement some basic procedures and show how they

could be used in a sorting algorithm:

The MAX-HEAPIFY procedure, which runs in O(lg n) time, is the key to maintaining

the max-heap property.

The BUILD-MAX-HEAP procedure, which runs in linear time, produces a max-heap from

an unordered input array.

• The HEAPSORT procedure, which runs in O(n lg n) time, sorts an array in place.

The MAX-HEAP-INSERT, and HEAP-EXTRACT-MAX procedures, which run in O(lg n) time, allow the heap data structure to implement a priority queue.

* **Sorting Techniques**

You’re required to compare the running time performance of your algorithms against:

– An O(n2) sorting algorithm such as Selection Sort, Bubble Sort, or Insertion sort.

– An O(n lg n) sorting algorithm such as Merge Sort or Quick sort algorithm in the average case.

You’re required to compare the running time performance of your algorithms against:

– An O(n2) sorting algorithm such as Selection Sort, Bubble Sort, or Insertion sort.

– An O(n lg n) sorting algorithm such as Merge Sort or Quick sort algorithm in the average case.

# UML Diagram

Diagram

Description automatically generated

Diagram

Description automatically generated

# 3.Analysis

* Analysis was performed on 5 batches (10, 100, 1000, 10000, 100000) nodes.
* The time for every batch of nodes was estimated 10 times and the average value was calculated at the end.

**Analysis Table:**

**Table

Description automatically generated with medium confidence**

**Charts:**

**Chart, line chart

Description automatically generated**

Chart

Description automatically generated