

Priority Queue Sort Experiment

Holger Dell, Martin Aumüller, Riko Jacob

Finish by: September 14, 2020

Format of Hand-in This exercise can be solved in groups of up to three people. Please limit your report to 1 page of text (figures and graphs don't count).

This exercise deals with two ways to implement sorting.

Definition 1 (Sorting) *Given a list of numbers A of size n and an ordering defined by an operator \leq , permute the list to give a result B such that if $A[i] \leq A[j]$, then $A[i]$ appears before $A[j]$ in B .¹*

Your task in this assignment is to construct an experiment comparing two methodologies:

1. An implementation that sorts using a library sorting function.
2. An implementation that sorts using a library priority queue.

The “priority queue” should at least support functions for **Insert** (which inserts an item into the data structure) and **DeleteMin** (which returns the smallest item in the data structure, while removing it from the data structure).² If you find a library data structure named “priority queue” it is probably fine for our definition; feel free to contact us or an TA if you have any specific questions. Note that each language may have different names for both the priority queue and the two operations we need.

An example of a library sorting function is `java.util.Collections.sort()` and an example of a library priority queue is `java.util.PriorityQueue`. Also in the `algs4` library, you can find several sorting functions and a priority queue.

Task: Design an experiment to compare these two methods, clearly stating the experimental design according to the concepts we discussed in class—your performance metric and indicator, hypothesis/question, parameters, type of experiment, results. Precisely specify the inputs considered in the experiments, and make sure to use several different classes of inputs (e.g., random, sorted, many equal keys) and their data types (e.g., integer, string). Give a plot of your data, and explain how the plot relates to your conclusions.

As a bonus question: can you find an example where the library priority queue outperforms the library sorting function?

¹If you're sorting integers then how to use \leq is fairly obvious. But this may change slightly based on your data: you can sort words in alphabetical order (for unicode strings, this might even depend on your locale), or floating point numbers using a floating-point-safe comparator, for example.

²Some priority queues will have **FindMin** and **Delete** as separate functions; this is fine for our definition.