# Priority Queue Sort Experiment

Sam McCauley

September 18, 2017

This assignment 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 me or the TA if you have any specific questions.

An example of a library sorting function is `java.util.Collections.sort()` and an example of a library priority queue is `java.util.PriorityQueue`.

**Assignment:** 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, parameters, type of experiment, results. Give a plot of your data, and explain how the plot relates to your conclusions. We will give feedback on how well this matches our expectations for experimental design on the exam.

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

[1] If you're sorting integers then how to use $\leq$ is fairly obvious. But this may change a bit based on your data: you can sort words in alphabetical order, or floating point numbers using a floating-point-safe comparator, for example.

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