



UPPSALA UNIVERSITET

Algorithms and Data Structures I 1DL210 Assignment 2

Daniel Hjelm
Daniel.Hjelm.0958@student.uu.se

Emanuel Wreeby
Emanuel.Wreeby.1646@student.uu.se

October 6, 2021

1 Comparison of the sorting algorithms

There are several differences between Insertion sort, Quick sort and Heap sort. Firstly, one difference is that Heap sort constructs a binary heap of the input array which the others don't. Secondly, Quick sort and Heap sort is built upon recursion due to their recursive calls within the algorithm. Also, Quick sort and Heap sort, consist of several building blocks (functions) with Quick sort having *Partition* and *QuickSort* as functions and Heap sort having *MaxHeapify*, *BuildMaxHeap* and *HeapSort*. Moreover, Quick sort and Insertions sort perform almost no unnecessary swaps whereas Heap sort always performs all swaps, even if all of your data is already sorted.

A situation where Insertion sort is better than the others in the best case when the array is already (or nearly) sorted or all the elements in the array is equal because of the running time being only $\mathcal{O}(n)$ for Insertion sort while it is $\mathcal{O}(n \log(n))$ for Heap sort and $\mathcal{O}(n^2)$ for Quick sort. An advantage that Heap sort have over the others is that the algorithm has a $\mathcal{O}(n \log(n))$ time complexity, even in the worst case. Consequently, one can always count on Heap sort being $\mathcal{O}(n \log(n))$ while in the worst case the others only perform $\mathcal{O}(n^2)$. Quick sorts advantage lies in it is efficiency and speed dealing with array sizes of smaller and moderate size. In addition, Quick sort sorts in place which means it does not need any additional storage.