Assignment 2

Albert Bjelvenmark, Svante Sundberg

Insertion sort:
Advantages with using insertion sort is that it does not change the order of elements with the same value. It is also adaptive, which means that if the input array is partly sorted, the algorithm will be more effective using a lesser amount of operations. The algorithm is also easy to implement and does not use much memory space. However, the best case running time for Insertionsort is O(n^2) which makes it slow for large inputs.

Quicksort:
Quicksort is a divide-and-conquer algorithm and with an average and best performance of O(nlogn) it is an effective sorting method. However in special cases it performs at a worst case at O(n^2) which is not favorable.

Heapsort:
Heapsort runs in O(nlogn) for both the worst and best case. The worst case scenario is therefore better than Quicksort but the best case the same.

Conclusion:
Quicksort and Heapsort algorithms are more efficient than Insertion sort when the input is very large. Insertion sort on the other hand could be useful when the input is smaller. When you are not really minding a slightly longer sorting time but instead is more concerned about memory.

Heapsort is usually slower than quicksort in practice, even though they both run in O(nlogn) in the average case, but for the worst case scenario Quicksort runs in O(n^2). If the input is already or almost completely sorted, Heapsort is preferred over Quicksort. This however can be avoided depending on how we choose the pivot element for Quicksort.