1DL210: Assignment #2

Due on Oct.6th

Algorithm and Data Structure I

Group 10 - Melat Getachew Haile, Tianren Sun, Xinyao Sun, Ziqi Kang

1 Comparison

1. What are the differences between the three algorithms, Insertion Sort, Quicksort and Heapsort?

	$Best-Case$	$Worst-Case$	$Average-Case$
$InsertionSort$	$\theta(n)$	$\theta(n^2)$	$\theta(n^2)$
$QuickSort$	$\theta(nlog(n))$	$\theta(n^2)$	$\theta(nlog(n))$
$HeapSort$	$\theta(nlog(n))$	$\theta(nlog(n))$	$\theta(nlog(n))$

We can see from the table above that different algorithms have different time complexity.
insertion-sort: Insertion sort algorithm pass the ordered sequence, for the data that has not been sorted, scan from the rear perspective in the sorted, find the corresponding position and insert it.
quick-sort: Divide a list into two sub sequences, and then sort the two sub sequences recursively.
heap-sort: In ascending order, the array is converted to the Max-Heap Heap. Repeatedly take out the node with the largest value from the largest heap (exchange the root node with the last node, and move the last node after the exchange out of the heap), and let the remaining heap maintain the maximum heap properties.

In order to achieve QuickSort and HeapSort, recursion is implemented. However, we don't need recursion in the Insertion Sort. Divide and conquer strategy is employed in QuickSort and HeapSort. On the other hand, this strategy is not necessary in InsertionSort.

2. For each algorithm, mention a situation where it has an advantage over the other two.

In the best situation, insertion sort has minimal time complexity - $\mathcal{O}(n)$.
In-place version of quicksort has a lowest space complexity of $\mathcal{O}(logn)$.
In the worst situation, heap sort has the minimal time complexity - $\mathcal{O}(nlogn)$.