# Sorting Algorithms Research

## 1. Bubble sort

The bubble sorting algorithm is one of the most straightforward sorting algorithms. The logic was already known, as it was introduced in last semester's C programming course. I included it in this research because I was curious about its performance against the other sorting algorithms. The algorithm compares two adjacent elements, swapping them if they are in the wrong order. The algorithm passes through the array numerous times until no more swaps are made and the array is sorted. The bubble sort is a stable sorting algorithm with a time complexity of **O(n²)**. The worst case occurs when the array is reverse sorted, while a sorted array gives a big O value of O(n) for the best chance.

## 2. Selection sort

The selection sort algorithm sorts an array by constantly finding the minimum element from unsorted part and moving it to the beginning of the array. It is similar to the bubble sort, but in every iteration of selection sort, the smallest element from the unsorted subarray is picked and moved to the sorted subarray. The time complexity is also **O(n²)** because the algorithm uses two nested loops.

## 3. Merge sort

The merge sorting algorithm is clearly the most complex in this list of sorting algorithms. The array is split into two sub-arrays again and again until only one element is left. The algorithm has a recursive behaviour, as the **spiltArray()** function calls itself until the size of the array is 1. Afterwards all the sub-arrays are merged into a new array in a sorted order. Merge sorting is known to be generally efficient, since it has a time complexity of **O(nlogn)** in the best and worst case.

## Time performance comparison

The table below compares the different sorting algorithms performance with different array sizes. The time is recorded in seconds.

|  |  |  |  |
| --- | --- | --- | --- |
| Number of elements N | 10 | 1000 | 10000 |
| Bubble sort | 0.000001 | 0.001503 | 0.215817 |
| Selection sort | 0.000001 | 0.001063 | 0.102582 |
| Merge sort | 0.000002 | 0.000126 | 0.001356 |

## Conclusion

Merge sorting is slower comparative to the other sorting algorithms when the number of elements is small, as it goes through the whole process even if the array is sorted. It also requires an additional memory space of 0(n) for the temporary array. However, merge sorting performs a lot better compared to the other two sorting algorithms, when the number of elements in the array increases.

## References

<https://www.geeksforgeeks.org/bubble-sort/>

<https://www.tutorialspoint.com/data_structures_algorithms/selection_sort_algorithm.htm>

<https://www.youtube.com/watch?v=JSceec-wEyw&ab_channel=GeeksforGeeks>