*Lecture 11:*

Analyse!

Binary Search:

Look at slides for how it is done and pseudocode.

Time complexity is O (log2(n))

Sorting algorithms

A sorting algorithm is **stable** if it does not change the order of the elements that have the same value.

A sorting algorithm is **in-place** if the memory used is O(1).

Insertion sort

Slides for pseudocode and example.

It is **stable** and **in-place**.

Worst-case time complexity -> Big-Theta (n ^ 2)

{O(n^2) comparisons and swaps}

Average-case time complexity -> Big-Theta (n ^ 2)

{O(n ^ 2) comparisons and swaps}

Best-case time complexity -> Big-Theta (n)

{O(n) comparisons and O(1) swaps}

Overall insertion sort is good for sorting small arrays as it is stable (no unnecessary moves) and in-place (low memory requirement).

Selection sort

Slides for pseudocode and example.

Selection sort is **in-place** but **not stable**.

Worst-case, average-case and best-case performance O(n ^ 2) comparisons; O(n) swaps.

Selection sort is better if the swapping of elements is hard. (Because selection sort makes at most n-1 swaps)