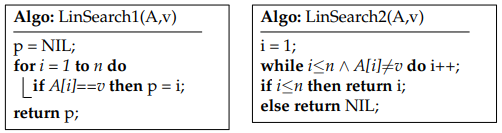
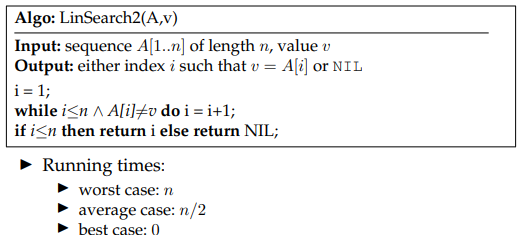
# Algorithms

**Linear search**:



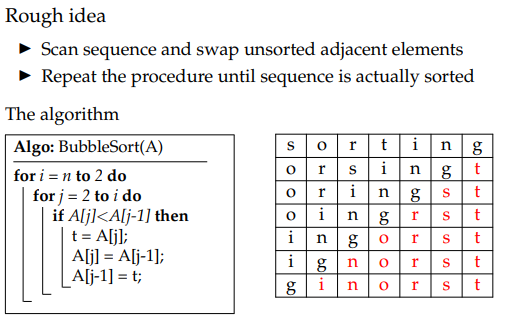
**Binary search**:

Sorted list entered:



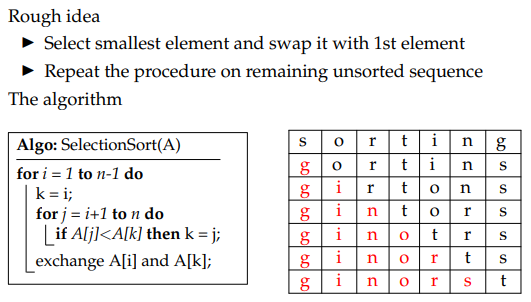
# Sorting

## Bubble sort



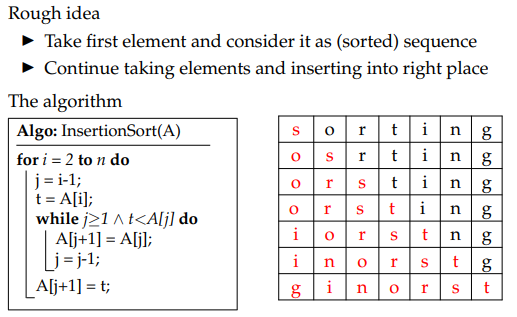
* Many comparisons
* Different amount of exchanges

## Selection sort



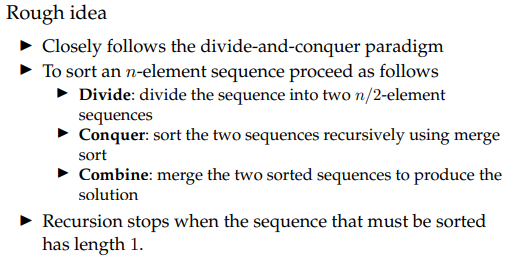
* Consistent number of comparisons and exchanges

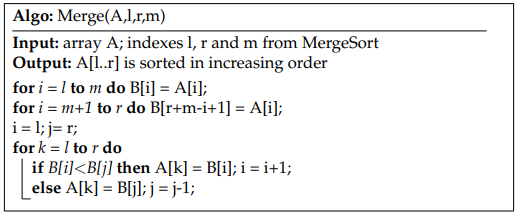
## Insertion sort



Inconsistent comparisons and exchanges

## Merge sort





## Heap sort

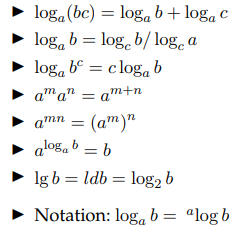
# Algorithmic complexity

# Correctness

An algorithm is correct if it terminates and produces the desired output for any legal input.

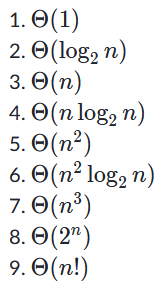
# Asymptotic complexity

Goal: simplify the analysis of running time



Big-O notation: worst case analysis

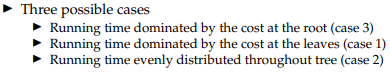
Big-Ω notation: best case analysis

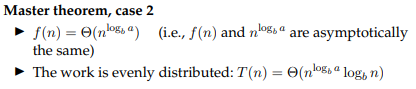


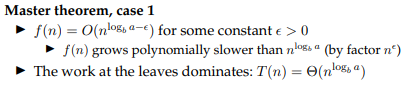
# Recurrences

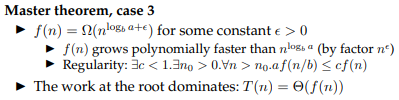
Master method

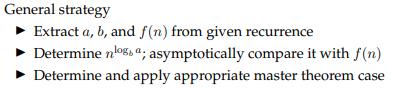






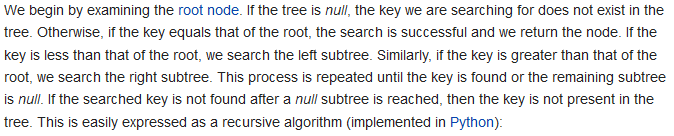




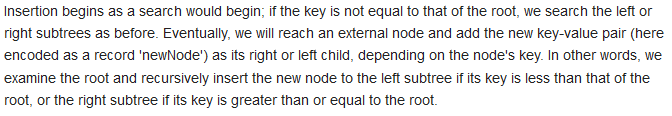


# Binary trees

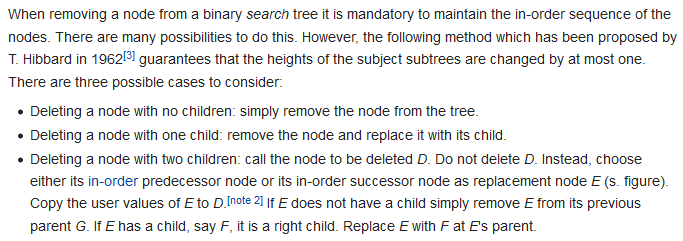
Searching:

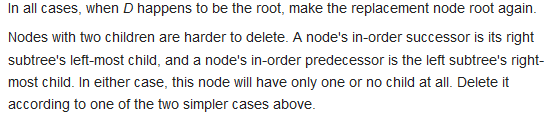


Insertion:



Deletion:





Predecessor: rightmost element in the left subtree

Successor: leftmost element in the right subtree

## Quicksort

Best case: n\*log n

Worst case n^2

# Red-black tree

