First Steps: Search and Performance

Learning Goals:

- You can apply a binary search for a list of data
- You can estimate the number of steps it takes to find the result (number of operations)

1 A simple search algorithm: Binary Search

Imagine you have to search for a name in a telephone book. How would you proceed? What is the most efficient way to do a search? Consider the hand-out paper *running_time_algorithms* to give you an idea how to search.

- ➔ Now implement a binary search with a primitive array (an array with a fixed size of elements). Make sure you define the search space accordintly (index of the array from 0 to n-1).

2 Number of steps

If we do a simple search, the number of steps would be 100 in the worst case (if we have a list with 100 elements).

- ➔ How many steps will we have roughly if we use a binary search?

Make a rough estimate and then compare it with the results given in the hand-out.

As you can see, the number of steps in an algorithm are quite important. They indicate how quickly an algorithm will determine (ie. will come to an end).

The performance of an algorithm is crucial for efficient data processing. We will see later that there are different types of algorithm depending on their complexity. Binary search is of the type “O(log n)” or logarithmic complexity. But there are others as well (see a simple overview here: https://yourbasic.org/algorithms/big-o-notation-explained/) .