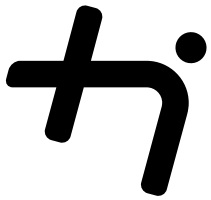
Technische Hochschule Ingolstadt
Prof. Dr.-Ing. Richard Membarth



12. April 2023

Algorithms and Data Structures
Assignment 4

Submission deadline for the exercises: 16. April 2023

4.1 Search

In the following, we want to compare the performance of binary search and interpolation search.

a) In the lecture, we learned about the recursive implementation of binary search. Here we will look at an iterative implementation. Implement the function `binary_search()` provided in the `search.py` file. The implementation should be iterative and return the index if the key was found, otherwise -1.

b) For interpolation search, implement the function `interpolation_search()` provided in the `search.py` file. Interpolation search works similar to binary search, but computes mid according to the following formula:

$$mid = low + \left\lfloor \frac{(key - arr[low]) \cdot (high - low)}{arr[high] - arr[low]} \right\rfloor$$

The implementation should be iterative and return the index if the key was found, otherwise -1. Note that the termination check for interpolation search needs to be extended to also check the key against the lower and upper array elements.

c) For benchmarking, a reference implementation for `linear_search()` is also provided in `search.py`. Compare the runtime of linear search, binary search, and interpolation search for an array with 1 million elements for different keys. For initialization of the array with random data, you can use the following code:

```
1 import random
2
3 arr = [random.randint(0, 10000000) for i in range(1000000)]
4 arr.sort()
5 keys = [arr[random.randint(0, 1000000)] for i in range(10)]
6 keys.extend(random.randint(0, 1000000)  for i in range(5))
```