# Assignment 1 – Search in dynamic sets

There are a number of algorithms designed to efficiently search for elements in dynamic sets: binary search trees, multiple approaches to balancing them, hash tables, and multiple approaches to resolving collisions. Different algorithms are suitable for different situations according to the nature of the processed data, distribution of values, performed operations, etc. In this task, your task is to implement and compare these approaches.

Your task in this assignment is to implement and then compare 4 implementations of data structures in terms of the effectiveness of **insert**, **delete** and **search** operations in different situations:

- (3 points) Implementation of a binary search tree (BVS) with any balancing algorithm, e.g. AVL, Red-Black Trees, (2,3) Trees, (2,3,4) Trees, Splay Trees, ...
- (3 points) Second implementation of BVS with a different balancing algorithm than in the previous point.
- (3 points) Implementation of a hash table with collision resolution of your choice. The hash table size adjustment must also be implemented.
- (3 points) Second implementation of the hash table with collision resolution in a different way than in the previous point. The hash table size adjustment must also be implemented.

You can obtain a total of 12 points for the implementations themselves according to the points above. You should upload each implementation in one separate source file (if you want to upload all four, you upload them in four files). **It is not allowed to download foreign source code! You must implement at least two of the above implementations for a successful upload.** Verify correctness by testing - comparison with other implementations.

In the technical documentation, your task is to document all implemented data structures and provide detailed testing scenarios, based on which you have found out in which situations which of these implementations are more effective. **It is also required to submit a program that is used to test** and measure the effectiveness of these implementations as a single source file (it contains the main function). **Without a test program, and thus without a successful comparison of at least two implementations, the solution will be evaluated with 0 points.** You can get a maximum of 8 points for documentation, testing and achieved results (identifying suitable test scenarios). The quality of testing and processing of results in the documentation is mainly evaluated. Tables and graphs with results that compare the performance (speed) of individual solutions should be included as well. As the results depend not only on the implementation of the solution, but also on the test scenarios (sequence of insert, delete and search operations), it is important to try and document various different test scenarios.

You can obtain a total of 20 points at most. The minimum requirement is 8 points.

The solution of the assignment must be uploaded in AIS by the set deadline (delays are allowed only in very serious cases, such as sickness, the decision whether the solution is accepted after deadline is decided by the instructor). One zip archive should be uploaded, which contains individual source code files with implementations, a test file and one documentation file in **pdf** format.