

**Course Name:** Advanced Data Structure and Algorithms  
**Course Code:** CS 507  
**Credit:** 3  
**Category:**  
**Prerequisites:**

**Course contents:**

Introduction to runtime analysis of algorithms, recap of sorting etc from course „Basic Datastructures“. RAM model, abstract runtime of algorithms. Growth of functions, upper, lower, exact bounds, working with asymptotic runtime behaviour.

Recap of linear data structures. Dictionaries as an interface, comparing linked lists with arrays. Dynamic arrays – a first example in amortized analysis. Skip lists.

Search trees. Balanced search trees: AVL trees, Splay trees, 2-3 trees, Red-black-trees. Search trees for block devices (disk et al.): B-trees, B+-trees.

Intro to multidimensional search trees: k-d-trees, R-Trees, [depending on time: reducing dimension – space filling curve (Hilbert curve)]

Hashing: basic algorithms, specific applications: consistent hashing, Distributed hash tables, Bloom filters, [Merkle trees]

[Tries: data structures for information retrieval ]

Graphs: DFS/BFS, Shortest path, minimal spanning tree.

**Literature:**

Cormen et al: *Introduction to Algorithms*, MIT press,

Skiena: *Algorithm Design handbook*, Springer Verlag

Weiss: *Data Structures and Problem Solving (Java or C++)*, Pearson / Addison Wesley

**Project:**

You are supposed to implement dictionaries( insert(), find() delete() ) in six different ways:

Binary search tree, AVL tree, Splay tree, 2-4 (also: 2-3-4)trees, Redblack trees, Hashing.

Furthermore a test bed which allows to compare them, is needed. You should devise a comparison methodology and write a report on that.