DAST – Final Lecture

Summary and overview

- What we have learned.
- Why it is important.
- What next.

What have we learned?

This course can be viewed as a thorough introduction to the main directions in Computer Science:

- Theoretical computer science
  algorithms design and rigorous analysis
- Programming and implementations
  Abstract data types, encapsulation, data structures
- The combination of the two

Topics

1. Tools for algorithm analysis
2. Sorting
3. ADTs and Data structures
4. Graph algorithms
5. Special topic: Huffman coding

Extensive hands-on JAVA programming

1. Tools for algorithm analysis

- Algorithm correctness
  - Loop invariants
  - Formally proving properties of algorithms
- Algorithms complexity analysis
  - Time and space complexity measures
  - Asymptotic complexity functions: O, Θ, Ω
  - Worst case, average case, best case
  - Upper and lower bounds
  - Amortized, expected.
  - Recurrence equations: how to formulate and solve them. The Master theorem.

2. Sorting

- Sorting is a basic operation of many algorithms!
- Comparison-based algorithms
  - Reviewed Insertion Sort, Merge-Sort, Bubble-sort
  - Quick-Sort and its complexity analysis
  - Randomization
  - Lower bound proof
  - Heap sort
- Non-comparison based algorithms
  - Counting sort
  - Radix sort
  - Bucket sort

3. ADTs and Data Structures (1)

- Ordered collections: insert, remove, first, last, find
- Stacks: push, pop, top
  - Array and Linked List implementations
- Queues: enqueue, dequeue, front
  - Circular array, Linked List
- Priority Queues: insert, min, delete-min, decrease-key
  - Binary Heap (array and binary tree implementation).
- Prefix-code tree: Binary tree

3. ADTs and Data Structures (2)

- Sorted Set: insert, remove, find, find-kth, min, max, successor, predecessor.
  - Balanced binary trees: Red-Black Trees, AVL trees, B-trees.
- Dictionaries and Tables: insert, remove, find
  - Hash tables: chaining and open addressing implementation.

3. ADTs and Data Structures (3)

- Graphs
  - Adjacency lists
  - Adjacency matrix
- Disjoint Sets: MakeSet, FindSet, Union
  - Linked lists
  - Trees

4. Graph algorithms

- Graph and shortest path properties
- Unweighted graph traversals: BFS, DFS
- Strongly Connected Components: SCC
- Minimum spanning tree algorithms: MST
  - Prim, Kruskal
- Weighted shortest path algorithms:
  - Dijkstra's shortest paths algorithm
  - Bellman-Ford shortest path algorithm
- All-shortest path algorithms
  - Floyd-Warshall algorithm; Transitive closure

5. Special topic: Huffman coding

- Prefix code properties
- Optimal coding
- Greedy algorithm for optima code construction

Important Computer Science concepts

- Mathematical tools for algorithm analysis.
- Proving algorithm correctness.
- Lower bounds: show that no algorithm can do better than a certain bound.
- Algorithmic paradigms:
  - Randomization
  - Heuristics
  - Greedy methods
  - Dynamic programming
- How to design and adapt an ADT.

ADTs and Data Structures

- The notion of an ADT: The interface.
- The notion of the data structure: algorithms and ways to implement the ADT efficiently.
- Preparation for more abstract design of algorithms you will see in the Algorithms course.

What next?

Core courses
- Algorithms A and B
- Computability

Elective courses
- Artificial Intelligence
- Computational geometry
- Cryptography
- …

בהצלחה!!