TDDB56 – DALG
Examination Requirements
To be augmented for TDDB57

Overview:
- Complexity Analysis [G&T] 4.2; Fö 1-2; Le 1
- Stacks and Queues [G&T] 5.1, 5.2; Fö 3; Le 2
- ADTMap&Dictionary, Hashing, Skip Lists [G&T] 9.1-9.4.1; Fö 4; Le 2
- Trees, Search Trees, [G&T] 7.1-7.3; 10.1-10.4; Fö 5-7; Le 3
- Priority Queues, Heaps, Union/Find [G&T] 8.1-8.3, 11.6.2; Fö 7; Le 4
- Sorting and Selection [G&T] 3.1.2, 11.1-11.5, 11.7; Fö 8-10; Le 4
- Graphs [G&T] 13.1-13.4; Fö 11; Le 5

[G&T]: Goodrich and Tamassia 4th edition
[Le]: on-line lektion’s material  [Fö]: on-line slides

Complexity Analysis:
- Order Notation: O, Θ, Ω
  - [G&T] 4.2.3-4.2.4;
  - Definitions and intuitions of asymptotic analysis;
- Analysis of algorithms, recurrence equations
  - [G&T] 4.25, 6.1.5, 11.1.5;
  - Finding complexity of given iterative/recursive algorithms
Cases: worst case, expected, amortized analysis

Stacks and Queues:
- Stack
  - ADT [G&T] 5.1.1
  - Implementations [G&T] 5.1.2, 5.1.3
  - Applications [G&T] 5.1.4
- Queue
  - ADT[G&T] 5.2.1
  - Implementations [G&T] 5.2.2, 5.2.3

Which of the algorithms discussed in the course use stacks or/and queues?

Trees:
- Basic Terminology [G&T] 7.1.1
- ADT Tree/binary Tree and its implementation
  - [G&T] 7.1.2 – 7.1.3, 7.2, 7.3.1, 7.3.3-7.3.6
- Binary Search trees [G&T] 10.1
Particular attention: representation of trees operations on BSTs

Maps and Dictionaries:
- Implementations ADT Map/Dictionary:
  - Simple list-based [G&T] 9.1.1, 9.3.1
  - Hash tables/collision handling (chaining, open addressing, double hashing)
    [G&T] 9.2.1-9.2.2, 9.2.4-9.2.5, 9.2.7, 9.3.2
  - Ordered Search Tables/Binary Search [G&T] 9.3.3
  - Skip Lists [G&T] 9.4.1
Illustrating execution of ADT operations in these implementations
**Special Search Trees:**
- AVL Trees  \[G&T\] 10.2
- Splay Trees  \[G&T\] 10.3.1-10.3.2
- Multi-way Search Trees, (a,b) trees  \[G&T\] 10.4
- B-trees  \[G&T\] 14.3.2

Examples showing execution of look-up, insert and delete operations on AVL-trees, Splay trees, (2,3) trees.

**Priority Queues**
- ADT Priority Queue  \[G&T\] 8.1
- Implementations:
  - Lists  \[G&T\] 8.2.1,8.2.2
  - Heap  \[G&T\] 8.3.1-8.3.3

Show execution of ADT Priority Queue operations in different implementations; variants of heaps.

**Sorting and Selection (1)**
- Aspects of Sorting  \[Fö\] 8.3-8.5
- Comparison based sorting algorithms
  - (Linear) Insertion Sort  \[G&T\] 3.1.2
  - (Straight) Selection Sort  \[G&T\] p.332
  - Quick Sort  \[G&T\] 11.2
  - Heap Sort  \[G&T\] 8.3.5-8.3.6
  - Merge Sort  \[G&T\] 11.1.1-11.1.3
- Lower bound  \[G&T\] 11.3

Illustrate steps of a sorting algorithm for given input, discuss stability of a given algorithm, discuss complexity.

**Graphs**
- ADT Graph  \[G&T\] 13.1
- Representing graphs
  - adjacency list  \[G&T\] 13.2.1-13.2.2
  - adjacency matrix  \[G&T\] 13.2.3
- Graph traversals and applications
  - Depth-first Search  \[G&T\] 13.3.1
  - Breadth-first Search  \[G&T\] 13.3.3
- Directed graphs
  - Basics/traversals  \[G&T\] 13.4.1
  - Strong connectivity  \[G&T\] 13.4.1
  - Topological ordering on DAGs  \[G&T\] 13.4.3

Show data structures representing a given graph, illustrate on examples basic graph algorithms.