Overview:
• Complexity Analysis [G&T] 4.2 ; Fö 1
• Sorting and Selection [G&T] 3.1.2,11.1-11.5,11.7; Fö 2
• Basic ADT’s Stacks,Queues,Lists,Trees [G&T] 5.1-5.3, 6.1-6.2, 7.1-7.3; Fö 3
• Graphs [G&T] 13.1-13.4; Fö 4
• ADT Map/ADT Dictionary, Hash Tables, Binary Search Trees [G&T] 9.1-9.3, 10.1-10.1.2; Fö 5
• Balanced Search Trees: AVL, multiway e.g. (2-3); [G&T] 10.2,10.4, Fö 6
• ADT Priority Queue, Heap, Heap Sort [G&T] 10.2,10.4, 14.3, 8.1, 8.3; Fö 6
• Union-Find Structures 11.6.3; Fö 7

[GT]: Goodrich, Tamassia 4th ed. [Fö n ]: on-line slides

Sorting and Selection (1)
• Aspects of Sorting [Fö] 2.4
• Comparison based sorting algorithms
  – Insertion Sort [Fö] 2.5-2.6 [G&T] p.333
  – Selection Sort [Fö] 2.7-2.8
  – Quick Sort [Fö] 2.9-2.13 [G&T] 11.2
  – Heap Sort [G&T] 8.3.5-8.3.6 [Fö] 6.27
• Lower bound [G&T] 11.3 [Fö] 2.15-2.16

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

Sorting and Selection (2)
• Bucket Sort [G&T] 11.4.1 [Fö] 2.17-2.18
• Radix Sort [G&T] 11.4.2 [Fö] 2.19-2.20
• Selection
  [G&T] 11.7 [Fö] 2.21-2.23

Show steps of a sorting/selection algorithm for given input, discuss complexity

Stacks and Queues:
• Stack
  – ADT [G&T] 5.1.1 [Fö] 3.5
  – Implementations [G&T] 5.1.2, 5.1.3 [Fö] 3.8
  – Applications [G&T] 5.1.4 [Fö] 3.6-3.7
• Queue
  – ADT[G&T] 5.2.1 [Fö] 3.9
  – Implementations [G&T] 5.2.2, 5.2.3 [Fö] 3.10-3.12

Show implementation of ADT operations; for example circular array implementation of ADT Queue
Trees:

• Basic Terminology [G&T] 7.1.1 [Fö] 3.15-3.16
• ADT Tree/binary Tree and its implementation [G&T] 7.1.2-7.1.3, 7.3.1, 7.3.3-7.3.6 [Fö] 3.17-3.20
• Tree traversals [G&T] 7.2 [Fö] 3.21-3.22

Particular attention:
- representation of trees
- tree traversals

Graphs

• ADT Graph [G&T] 13.1 [Fö] 4.3-4.5
• Representing graphs
  – adjacency list [G&T] 13.2.1-13.2.2 [Fö] 4.6-4.7
  – adjacency matrix [G&T] 13.2.3 [Fö] 4.6-4.8
• Graph traversals and applications
  – Breath-first Search [G&T] 13.3.3 [Fö] 4.14-4.18
• Directed graphs
  – Topological ordering on DAGs [G&T] 13.4.3 [Fö] 4.21-4.22

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

Maps and Dictionaries:

• ADTMap, ADT Dictionary
  [G&T] 11.6.1-1.9.1, 9.3 [Fö] 5.3-5.4,
• Implementations ADT Map/Dictionary:
  – Simple list-based [G&T] 9.1.1, 9.3.1
  – Hash tables/collision handling (chaining, open addr,...) [G&T] 9.2.1-9.2.2,9.2.4-9.2.5, 9.2.7, 9.3.2 [Fö] 5.8-5.19
  – Ordered Search Tables/()inary Search [G&T] 9.3.3 [Fö] 5.6-5.7

show execution of ADT operations in these implementations;

Balanced Search Trees:

• AVL Trees
  [G&T] 10.2 [Fö] 6.3-6.10
• Multi-way Search Trees (specifically (2,3)-trees)
• B-trees
  [G&T] 14.3.2 [Fö] 6.21

examples showing execution of look-up, insert and delete operations on these trees

Priority Queues

• ADT Priority Queue
• Implementations:
  – Lists [G&T] 8.2.1,8.2.2
  – Heap [G&T] 8.3.1-8.3.3 [Fö] 6.23-6.26

show execution of ADT Priority Queue operations on heap

Union-Find Structures

• ADT Partition (Union-Find)
  [G&T] 11.6 [Fö] 7.23
• Implementations:
  – Lists [G&T] 11.6.2

show execution of the operations of ADT Partition on trees