CMCS420 Midterm 1 Prep

Midterm 1 will be in class, March 8th, and 75 minutes long. You will be allowed a one sheet of notes, front and back.

Questions can cover topics up to and including skip lists. Those topics covered in homework and projects will be given more emphasis than skip lists and treaps, though, since you haven’t had solved problems on them. Questions will come from the homework; issues that arose in the projects; practice questions posted by Mount; additional examples posted soon by Eastman; and learning aims from each lecture. Mount’s practice midterm is at:

https://www.cs.umd.edu/class/spring2022/cmsc420-0101/handouts.html

Hint: practice and actual midterms can be found on the course pages for previous semesters. Older midterms may have splay trees which you are not responsible for – in general, unless we’ve covered the material, you’re not responsible for it. Here’s spring 2021 – you can find links to others at https://www.cs.umd.edu/class/

https://www.cs.umd.edu/class/spring2021/cmsc420-0101/

The daily lecture handouts give learning aims and readings. Eg, after this lecture you should be able to. The midterm is likely to be structured like the practice questions posted by Mount – short answer that ask for details on particular data structures; questions that ask for you to draw or otherwise represent data structures and their operations; coding questions; and analysis questions.

A review of lecture learning aims

I. Lecture 1: introduction
   a. Distinguish abstract from concrete data structures
   b. List some questions to ask when given a data problem to solve

II. Lecture 2: Lists and the linear world
   a. Describe how lists can be specialized for many ADT
   b. Describe and apply different concrete representations of linear ADTs
   c. Give details of a concrete to abstract representation mapping when needed
   d. Identify the expensive or complicated operation in a concrete implementation
   e. Recognize and apply amortization or charging arguments for expensive operations

III. Lecture 3: Rooted and binary trees
   a. Use standard graph and tree terminology
   b. Compute standard measures of tree – depth, height, etc.
   c. Discuss and use concrete tree representations
      Raw graph, ordered traversals, array, binary (for binary and n-ary), threaded, etc
   d. Convert between representations – traverse a tree, reconstruct from traversals, etc

IV. Lecture 4: Binary trees
   a. Use standard graph and tree terminology

Disclaimer: This was prepared for Prof. Eastman's section. Some material may not apply to Mount's section.
b. List, explain and analyze the cost of map operations in ordered and unordered arrays
c. Explain the representation of a map in an unbalanced binary search tree
d. Explain, pseudocode, code and analyze find, insert and delete in a binary tree

V. Lecture 5: AVL trees
a. Explain the what, why and how of AVL trees
b. Give and apply the definition of AVL trees as height balanced
c. Demonstrate and work with the argument that AVL trees are $O(\log n)$
d. Explain and apply rebalancing techniques for AVL insertions and deletions

VI. Lecture 6: 2-3 trees
a. Explain why you might use other than binary trees for search
b. Give the definition and properties of a 2-3 tree and identify one
c. Trace examples of the insert and delete operation in small trees

VII. Lecture 7: Red-black and AA trees
a. Use case analysis for data structure operations
b. Organize AVL tree cases well
c. Organize 2-3 tree inserts and deletes
d. Convert 2-3 trees to red-black trees
e. List conditions of red-black trees
f. Work with AA tree inserts (and maybe deletes)

VIII. Lecture 8: Treaps
a. Give a rough idea of the number of binary trees
b. Explain the randomization behind treaps
c. Perform insertion on a treap
d. Perform deletion on a treap

IX. Lecture 9: Deletions in binary trees
a. Work better with deletions in various forms of binary trees
b. Revisit priority queues in heaps
c. Could end up with forest of little trees. Need to merge now?

X. Lecture 10: Quake heaps
a. Explain the relative efficiencies of linked lists, binary heaps and quake heaps
b. Work with operations on Quake heaps
c. Explain the Quake heap metric for imbalance

XI. Skiplists (Mount's section)
(same items as for Treaps)