Data Structures and Algorithms

Final Exam Information

1. The final exam will be Wednesday, May 18 in Harney 235 (the Kudlick Classroom) at 8 am.
2. You’ll have three hours to complete the exam.
3. The exam is closed book and closed notes.
4. The exam is comprehensive: it may cover anything we’ve covered since the beginning of the semester.
5. There will be a somewhat greater emphasis on material covered since the second midterm: sorting and hashing.
6. My office hours during exam week are Monday, May 16 11am–1pm.
7. Topics covered before the first midterm.
   (a) ADT’s and data structures
   (b) Big oh, big omega, big theta
   (c) Using big theta to predict a program’s performance
   (d) Caveats when predicting performance
   (e) Formula \( \log_c(x) = \frac{\log_b(x)}{\log_b(c)} \)
   (f) By hands proof that \( T(n) \) is \( O(f(n)) \) or \( \Omega(f(n)) \).
   (g) Program 1: why use a dummy header? why use a free list?
   (h) List ADT: array, singly-linked, doubly-linked
   (i) Program 2: heap implementation
   (j) Stacks, elimination of recursion
   (k) Queue, array implementation
   (l) Binary trees, terminology, traversals, implementations
   (m) Expression trees, postfix expression evaluation
   (n) Binary tries
   (o) Theorems regarding number of leaves and number of internal nodes
   (p) Storage requirements and overhead for binary trees
   (q) Binary search trees: insert, remove, member
   (r) Performance of BST ops.
(s) Roughly speaking, the first midterm covered chapters 1, 3, 4, 5.1–5.3, 5.5, and programming assignments 1 and 2.

8. Topics after the first midterm and before the second midterm

- More on run time of code and algorithms: big-$O$, big-$\Omega$, big-$\Theta$.
- Proof using mathematical induction.
- B-Trees: insert, delete, member
- Runtime of B-tree ops.
- Priority queues: complete, partially-ordered binary tree, min-heap and max-heap implementations. Performance.
- Implementation and runtime of buildHeap.
- Graphs and digraphs: definitions, implementations.
- DFS, BFS, topological sort.
- Strong components algorithm. Reduced graph of a digraph.
- Dijkstra’s algorithm: implementations, runtimes.
- Spanning trees, min-cost spanning tree.
- Prim’s algorithm: implementations, runtimes, correctness.
- Roughly: Sections 2.6.2, 3.4–3.5, 5.6, 7.1–7.3, 7.4.1, 7.5.1, and B-trees and Strong components.

9. Topics covered after the second midterm

(a) Bubble sort, selection sort, and insertion sort. Performance.
(b) Shellsort, increments, performance.
(c) Quicksort, implementation details, performance.
(d) Mergesort, heapsort. Performance.
(e) Bucket sort.
(f) Comparison-based sorting algorithms are $\Omega(n \log(n))$.
(g) Hash tables: open hashing, bucket hashing, "classical" closed hashing.
(h) Properties of good hash functions.
(i) Performance of hashing ops.
(j) Roughly: Sections 8.1–8.6, 8.9, 10.4.