Sample final questions

March 18, 2013

Good ways to study:

1. Go over solutions to homeworks
2. Go over class notes
3. Read the book
4. Study with your reading group

Basics:

1. Proofs by induction
2. Find loop invariant and prove correctness via the Initialization, Maintenance and Termination parts.
3. Divide and conquer and how it leads to recurrences
4. O, Omega, Theta, o and omega notations

Questions focusing more on second half of class:

1. All the advantages of Radix Sort How do you sort numbers in the range from 0..nk in O(n) time for any constant k?
2. Hashing by chaining and open addressing.
3. Linear & quadratic probing, double hashing.
4. Examples of good hash functions: Explain the multiplication method at a high level
5. BSTs, search, insert, delete. Worst-case and average case performance for various operations.

6. Worst-case and average case for building a BST via repeated insert ops

7. What is the relationship between QuickSort and BST’s

8. RBTs: What ops are efficient? How does insert work?

9. What are the advantages and disadvantages of various data structures?

10. Augmented data structures for implementing more ops efficiently?

11. BHs: What ops? How are they implemented?

12. How can you use other $O(\log n)$ operations to implement delete on a BH in $O(\log n)$ time?

13. Data structures for disjoint sets: what heuristics have what effect? What two key operations lead to dramatic improvements

14. Simple examples of Dynamic programming such as computing the Fibonacci Numbers efficiently. Why is the recursive alg inefficient?

15. Longest Common Subsequence Problem

16. Memoization: How to use Hashing for Memoization?

17. How to retrieve the solution for a dynamic programming problem

18. What are the advantages of the two representations of graphs?

19. Run basic algs: BSF, DFS, BH operations, Kruskal’s alg.

20. Running times of BFS and DFS, Kruskal’s alg

Chapters covered by final:
- Chapter 1-14 except 5, 9.2, 11.3.3, 11.5 and 14.3
- 15.1, 15.3, 15.4
- Binomial heaps (posted on class web page)
- 21.1-21.3
- 22.1-22.4
- 23.1, 23.2 Kruskal’s algorithm with Union Find