Exam 2 study guide

Master’s; Chapters 5-7, 12, 15. No graph search, MST etc (no Chapter 22 and beyond)

Builds up on previous topics, so familiarity on previous topics is essential!

Topics (not an exhaustive list)

~~Algorithm walk-thru and complexity~~

~~Heap – representation; MaxHeapify, BuildMaxHeap, etc~~

~~Heapsort; Quicksort, RandomizedQuicksort (and Partition)~~

~~Binary Search Trees (BST) –various orders, search, max/min, successor, predecessor, insert, delete, other primitives~~

Dynamic programming – steps, comparison, problems such as scheduling, MCM, N choose k, largest subsum, etc

~~Be able to give pseudocode for problems, derive their complexity~~

~~Strassen~~; solving recurrences –Master’s (I will give Master’s cases so no need to memorize)

Cheat sheet 1 letter size front-and back hand-written or computer generated.

**Sample problems**

Some short answers and some problems, similar to Exam 1 format. Like Exam 1, our (4620) student(s) gets some points even if 100% of the points are lost for each question☺

1. Short answers

DP: we normally solve each problem only once: True

Name one problem where DP was better than another solution: Rod cutting

MCM: what is the objective of this algorithm: computes the product of the matrix, in the fewest number of steps.

Algorithm for searching BST: What is its complexity? O(logn)

What is a balanced BST? a binary tree in which the height of the left and right subtree of any node differ by not more than 1, each side is similar, there isn’t a large incline on one side versus the other.

Can a heap be stored as an array? Yes,

What is the worst-case complexity of quick sort? N^2

How is binary search tree different than binary tree? The search tree is a sorted tree where each left child Is less than the parent and each right child is greater than.

2. Given a set of values draw the MaxHeap. OR BST

3. Quick sort

worst case(n^2)

average(nlogn)

best?

5. Write code for Fibonacci using DP.

6. Rod cutting: Where does DP come into play?

7. Design an algorithm for a given problem. Determine its complexity. Or, given an algorithm determine its complexity.

8. Given a recurrence relation, use Master’s.

Example: T(n) = T(2n/4) + 5

Need help with the masters’ theorem calculations, 3 cases.

9. Strassen: How is it different from traditional matrix mul algorithm? How did you figure out Strassen’s complexity?

Extra: