

Midterm 1 Study Sheet

Emphasis is towards Dynamic Programming.

1. Recurrences
 - a. Psuedocode implementation (top down and bottom up).
 - b. Derivation of recurrences from first principles.
 - c. Application of the Master Method (provided on test).
2. $T(n)$ [Worst or Average Case as appropriate] for various types of sorting.
 - a. Insertion Sort
 - b. Merge Sort
 - c. Heap Sort
 - d. Quick Sort
 - e. Counting Sort
 - f. Algorithmic Analysis in general
3. Binary Trees
 - a. Definition and use
 - b. Optimization utilizing dynamic programming
4. Assembly Line Problem, Matrix Multiplication
5. Memoization

Points to consider:

1. Difference between performing cost calculations of an algorithm $T(n)$, and cost calculations associated with an algorithm (Ex: matrix mult. & opt. binary search tree).
2. Impact of Tree Structure on $T(n)$ for various algorithms that use trees.
3. Understanding why you have to use big O notation as opposed to θ .
4. Understanding how to split a problem up into smaller subtasks that are identical to the larger task. Adding the appropriate terms to this division process to complete a recurrence.
5. For remembering $T(n)$ dependencies, understand the algorithm's operation, i.e. is it dependent on "divide and conquer", explicit loops, or data structures? Does it involve comparisons?