Midterm #2 Topics
Spring 2005

The midterm will be on Thursday May 19. It is a closed-book exam, but you may use one 3-by-5 card of notes written in your own hand. Calculators will not be allowed.

You are responsible for chapters 1 through 9 and 15 in the book\(^1\) as well as section 28.2 and appendices A and B. The second midterm will be comprehensive in that I will ask some questions on the material covered before the first midterm, but the material covered since the first midterm will be emphasized. (Note that techniques from the first part of the course may be required to solve questions relating to the more recent material.) Some of the longer questions on the the second midterm are likely to involve induction, decision tree lower bounds, and dynamic programming. Here are the topics we have covered in lecture since the first midterm.

1. Building a heap in linear time
2. Strassen's matrix multiplication,
3. Average case analysis of Quicksort, internal versus external randomization
4. Using indicator random variables
5. Radix sort (and bucket sort)
6. Linear time selection (median of median algorithm)
7. Lower bounds on problems
8. Information theoretic (decision tree) lower bound technique and how to apply it
9. The $\Omega(n \log n)$ lower bound for comparison sorting and its relationship to Radix Sort
10. The adversary lower bound technique
11. Dynamic Programming - the 6-fold path
12. Dynamic Programming algorithms for particular problems:
   (a) the number of knight's-move phone numbers
   (b) maximizing a skateboarder's excitement (largest value paths in a DAG)
   (c) parenthesizing a chain of matrix multiplications
   (d) computing the longest common subsequence
   (e) 0-1 Knapsack problem
   (f) Coin changing (homework)
   (g) Canoe rentals (homework)
   (h) Number of binary search trees (homework)

\(^1\)Chapter 5 is less important as long as you understand indicator random variables and the average case analysis of Quicksort, and much of chapters 6-8 should be review of material from CMPS 101.
Here are some additional study problems from the text, I don’t think I will have time to write up full solutions, but if you want to see how to solve particular problems I can outline solutions on WebCT.

1. Exercise 6.3-2 on page 135
2. Exercise 7-2.2 on page 153
3. Quicksort has been called an “in-place” sorting algorithm as it appears to take only a constant amount of space in addition to the array holding the elements to be sorted. How much space (in the worst case) is required for the “recursion stack” where the local variables for the Quicksort procedure are stored while recursive calls are being executed? How much space is required in the best case? (see Problem 7-4 on page 162 for a related series of questions.
4. Exercises 5.2-4 and 5.2-5 on page 99 (indicator random variables)
5. Show that there is no comparison based algorithm whose running time is linear for at least half the n! inputs of length n.
6. Exercise 8.3-3 on page 173
7. Problem 8-6 on page 180
8. Exercise 9.3-6 on page 192
9. Exercise 15.2-1
10. Exercise 15.3-2
11. Exercise 15.4-1
12. Exercise 15.4-5
13. Problem 15.6 on page 368