CSOR 4231 Midterm Exam
November 5, 2015, 4:10PM

Rules; Answer each question completely and concisely. When you give an algorithm, be sure to give the most efficient one you can, to prove that it is correct, and to analyze its running time.

Problem 1. [**10 POINTS**] Solve the following recurrences by giving big-O upper bounds. Justify the solution, either by the master theorem, or some other method learned in class.

1. [**5 POINTS**] $T(n) = 4T(n/2) + n^3$
2. [**5 POINTS**] $T(n) = T(n/2) + \log^2 n$

Problem 2. [**10 POINTS**] True or false. Briefly explain your answer.

1. [**4 POINTS**] $f(n) + g(n) = O(\max(f(n), g(n)))$.
2. [**2 POINTS**] All sorting algorithms take $\Omega(n \log n)$ time.
3. [**2 POINTS**] Multiplying 2 n by n matrices takes $\Omega(n^2)$ time.
4. [**2 POINTS**] Quicksort, with median-of-3 partitioning (as implemented in the homework problem), is always faster than Quicksort with a random partition.

Problem 3. [**20 POINTS**] You work in a chocloate factory. You need to back boxes of chocolates, with a goal of having each box have as close as possible to 500 grams of chocolate. More formally, you are given a series of n pieces of chocolate, numbered $1, \ldots, n$, where piece i has weight w_i (in grams). You must pack the chocolate into boxes, with the constraint that each box must contain consecutive chocolates, $i, i+1, i+2, \ldots, j$ for some i, j, with $i < j$. You are payed, based on how close each box is to 500 grams, that is, for a box with total weight $W = \sum_{k=i}^{j} w_k$, you earn $1000 - |W - 500|$ (you can be payed a negative amount). You total payment is the sum of the payments you receive from all your boxes. Your goal is to maximize your total payment.

Example: $w = (200, 350, 350, 100, 300)$. The optimal solution is to pack boxes $(200, 350), (350), (100, 300)$. Your total payment is $(1000 - |550 - 500|) + (1000 - |350 - 500|) + (1000 - |400 - 500|) = 950 + 850 + 900 = 2700$

This problem can be solved by dynamic programming. Prove that the problem has optimal substructure, give a recurrence for computing the optimal solution and explain what the running time of the resulting algorithm will be. You do not need to give pseudocode.

Problem 4. [**10 POINTS**] Prof. Random decides to assign grades the following way. There are are n students in the class, each with a unique id between

1 and n. The possible grades are the integers 1 through 10 and each grade is given to exactly $n/10$ students, chosen randomly. You can assume that n is a multiple of 10.

1. [**5 POINTS**] What is the expected number of students whose grade is identical to the the last digit in their id? Show your calculations.
2. [**5 POINTS**] What is the expected number of students whose grade is higher than the last digit in their id? Show your calculations.