CMPSCI 611: Advanced Algorithms
Micah Adler

Problem Set 1

Out: September 17, 2002
Due: September 24, 2002

Notes: *On homework assignments, you are allowed to discuss the questions with a small number of other people in the course. However, the emphasis of such discussions should be obtaining a solid understanding of the solutions to the assigned problem. Thus, you must destroy any notes from your discussions, and then write up the solutions on your own. For each problem, you must also list anyone you discussed that problem with (even briefly). You also must describe any other references you used.*

The homeworks are due at the beginning of class on the due date. Late submissions will be accepted only with special permission. Also, please take the time to write clear and concise answers. Credit will be reduced if answers are very unclear or long winded.

All questions count for the same amount of credit, although some are harder than others. Some of the questions may require quite a bit of thought, so I encourage you to start early.

1. a) How would you modify Strassen's algorithm to multiply $n \times n$ matrices in which n is not an exact multiple of 2? Show that the resulting algorithm runs in time $\Theta(n^{\log 7})$.

   b) Say you are given an algorithm that multiplies two $n \times n$ matrices in time n^α, for $\alpha < 3$. Using this algorithm, how quickly can you multiply a $kn \times n$ matrix by an $n \times kn$ matrix? Answer the same question with the order of the input matrices reversed.

2. In this question, we shall obtain a more exact bound on the running time of matrix multiplication, and use this to determine at what value of n Strassen's algorithm starts to outperform the naïve matrix multiplication algorithm. In order to do so, we shall use the fact that the solution to the recurrence relation

$$\begin{aligned} T(n) &= aT\left(\frac{n}{b}\right) + cn^\alpha \quad (n > 1); \\ T(1) &= d \end{aligned}$$

   is

$$T(n) = n^\beta \left(d + \frac{cb^\alpha}{a - b^\alpha}\right) - n^\alpha \left(\frac{cb^\alpha}{a - b^\alpha}\right),$$

   where $\beta = \log_b a$.

   a) Let $S(n)$ be the running time of Strassen's algorithm for multiplying two $n \times n$ matrices, where the entries in the matrix are integers, adding two integers requires 1 time unit, and multiplying two integers requires m time units.

   Give an exact recurrence relation (i.e., without any Θ or O expressions) for $S(n)$. Use the formula above to derive an explicit expression for $S(n)$ in terms of m and n. You should assume that n is a power of 2.

   b) Let $N(n)$ be the running time of the naïve matrix multiplication algorithm in the same model. Give an exact recurrence relation for $N(n)$, and derive an explicit expression for $N(n)$ in terms of m and n.

   c) For the case where $m = 10$, determine n_0, the smallest power of 2 such that $S(n) \leq N(n)$ for all $n > n_0$. Do the same for the case where $m = 1$.

3. [CLRS] problem 30-3.

4. The Acme Widget Factory is able to produce one widget per day. On the first day that the factory produces a widget, it has a set of customers $C = \{c_1, \ldots c_n\}$, where each customer c_i requests one widget by delivery date d_i, and has a price p_i. If customer c_i receives a widget produced on or before date d_i then it is *satisfied*, and pays p_i dollars; otherwise it is *unsatisfied* and pays nothing. The factory wants to assign the widgets it produces each day to the customers C in such a manner that the factory maximizes its profits. We can refer to the days as positive integers, where the first day is day 1, etc. Note that no customer is interested in more than one widget.

   a) Show that there is an optimal allocation of widgets where every customer is either satisfied, or does not receive a widget, and the satisfied customers receive their widgets in increasing order of their delivery dates.

   b) A set of customers $i \subseteq C$ is *feasible* if there is an allocation of widgets that satisfies every customer in i. Prove that $\{C, I\}$ is a subset system, where $I = \{i \text{ such that } i \subseteq C \text{ and } i \text{ is feasible}\}$. Prove that $\{C, I\}$ is a matroid.

   c) Describe a greedy algorithm for the problem of finding the optimal allocation of widgets. Show that your algorithm works correctly, and that it runs in $O(n^2)$ time.