Institut for Matematik og Datalogi
Syddansk Universitet

March 24, 2013
JFB

DM508 – Algorithms and Complexity – 2013
Lecture 1

Textbook and notes

Introduction to Algorithms, 3rd edition, by T. Cormen, C. Leiserson, R. Rivest, and C. Stein, MIT Press, 2009.

Extra notes (available in the bookstore): *DM508 Algoritmer og kompleksitet, Noter 4. kvartal 2013.* This is identical to the notes from 2010, 2011, and 2012 and is from the following sources:

- *Computer Algorithms: Introduction to Design and Analysis*, second edition, by S. Baase, Addison-Wesley, 1987.
- *Combinatorial Optimization: Algorithms and Complexity*, by C.H. Papadimitriou and K. Steiglitz, Prentice-Hall, 1982.

Format

Lectures and discussion sections will be in English. Please read the appropriate sections in the textbook or notes before coming to class and bring your textbook with you. There will both be assignments which you are required to turn in and other assignments which you should be prepared to discuss in the discussion sections (øvelserne), usually shortly after the relevant lecture. The lectures and discussion sections are scheduled to be in U20, except for the Monday lecture in week 17 and the Wednesday lecture in week 18, which will be in U15 and U24 respectively. The lectures will be on Mondays, 10:15–12 (weeks 15–20), on Tuesday, 12:15–14 (week 15), and on Wednesdays, 14:15–16 (weeks 16, 18, 20), but one of these will be cancelled. The discussion sections will be on Tuesdays, 12:15–14 (weeks 16–21) and Wednesdays, 14:15–16 (weeks 15, 17, 19 and 21), but one of these will also be cancelled. The "instruktor" for the course is Marie Christ.

The required assignments will be graded on a Pass/Fail basis, and satisfactory completion of all 3 assignments is required for a Pass. The assignments must all be turned in on time using the Blackboard system, submitted via the menu item "SDU Assignment". Turn in each assignment as a single PDF file. If you turn in a later, improved version, mark on the first page, near your name, which version it is (but we hope the system handles this). Keep the receipt it gives you proving that you turned your assignment in on time. You may work in groups of 2 to 3 students if you wish. These 3 assignments must be approved in order for you to take the oral exam, so cheating on these assignments is viewed as cheating on an exam. You are allowed to talk about course material with your fellow students, but working together on assignments with students not in your group is cheating. (You can, however, talk with Marie or me.) Using solutions you find elsewhere, such as on the Internet, is also cheating. You may do the assignments in either English or Danish, but if you write them by hand, please do so very neatly. You will be allowed to redo one of the first two assignments if it is not approved the first time (if one of your assignments is late, then you will have used up your only chance to redo any assignment).

The weekly notes and other information about the course are available through the World-wide Web. Use Blackboard or the URL:

http://www.imada.sdu.dk/~joan/dm508/

I have office hours 9:00–9:45 on Tuesdays and Thursdays.

There will be an oral exam on June 20 and 21, 2013. Previous exam questions are available on DM508's homepage; an updated set of exam questions will be available later in the course (probably the same as last year). You may do your exam in Danish if you wish (in most cases it is advisable to do it in Danish).

Lecture, April 8

We will begin with an introduction to the course. Lower bounds from section 2.4 of the first part of the notes will be discussed (part of this is also in section 8.1 of the textbook). We will also begin on sections 3.1 and 3.2 of those notes.

Lecture, April 9

We will finish sections 3.2, 3.3 and 3.5 of the DM508 notes, plus median finding from chapter 9 (section 9.3) in the textbook.

Problems to be discussed on April 10

Do problems:

1. Do problems 3.2 and 3.10 from pages 140 and 141 of the notes. For problem 3.2, use Stirling's approximation - formula 3.20 (2nd ed. 3.19) from the textbook, and compare your result to the upper bound for merging.

2. Prove a lower bound for merging two lists of lengths n and m which meets the upper bound of $n + m - 1$ (assume $n = m$).

3. From the following pages of that book by Baase:

Consider the problem of determining if a bit string of length n contains two consecutive zeros. The basic operation is to examine a position in the string to see if it is a 0 or a 1. For each $n = 2, 3, 4, 5$ either give an adversary strategy to force any algorithm to examine every bit, or give and algorithm that solves the problem by examining fewer than n bits.

and

a. You are given n keys and an integer k such that $1 \leq k \leq n$. Give an efficient algorithm to find *any one* of the k smallest keys. (For example, if $k = 3$, the algorithm may provide the first-, second- or third-smallest key. It need not know the exact rank of the key it outputs.) How many key comparisons does your algorithm do? (Hint: Don't look for something complicated. One insight gives a short, simple algorithm.)
b. Give a lower bound, as a function of n and k, on the number of comparisons needed to solve this problems.

4. From Baase's textbook: Suppose $L1$ and $L2$ are arrays, each with n keys sorted in ascending order.

   a. Devise an $O((\lg n)^2)$ algorithm (or better) to find the ith smallest of the $2n$ keys. (For simplicity, you may assume the keys are distinct.)
   b. Give a lower bound for this problem.

5. Design and analyze an efficient algorithm to find the third largest item in an array.

6. Consider the problem of Sorting by Reversals. You are given a permutation of the numbers from 1 to n in an array, A. The operation you have is to choose two indices, i and j, and to reverse the elements in the subarray from $A[i]$ to $A[j]$, inclusive. The objective is to end with a sorted array. For example, given $A = [8, 6, 4, 2, 7, 5, 3, 1]$, the first operation could be $(3, 6)$, resulting in $A = [8, 6, 5, 7, 2, 4, 3, 1]$. Then doing the operations $(2, 4), (3, 4), (5, 7), (5, 6), (1, 8)$ would finish sorting the array.

   a. Give an algorithm which sorts the array in $O(n)$ operations.
   b. Prove that any algorithm needs at least $\Omega(n)$ operations in the worst case.
   c. Why doesn't the information-theoretic lower bound of $\Omega(n \log n)$ apply here?

Assignment due Friday, April 19, 12:15

Note that this is part of your exam project, so it must be approved in order for you to take the exam in March, and you may not work with or get help from others not in your group (though you may talk with Marie Christ or myself). You may work in groups of two or three. You may write your solutions in English or Danish, but write very neatly if you do it by hand. Submit the assignment via Blackboard's "SDU Assignment" as one PDF file. Remember to keep a receipt. Turn in one assignment per group.

1. This problem is partially based on the game which in Danish is called "Sænke Slagskibe". In that game, two players each hide a number of ships (objects which have length k and width 1, where k usually varies between 2 and 5) on a rectangular grid, which we will say is m by n. A ship of length k will be placed in k consecutive coordinates in some row or some column. Ships cannot overlap – there cannot be two ships which are share any particular coordinate. Then the players take turns guessing coordinates (a letter and a number in the usual game, but we will assume guessing a row between 1 and m and a column between 1 and n. If a player names a coordinate containing part of a ship, the other player answer this that this is so. Let's call that a "hit". Otherwise, let's say that it was a "miss" and the player is informed of this. In the following $m = n = 5$, so the grid is 5×5. Write your lower bounds as adversary arguments.

   (a) Suppose first that there is only one ship and it has length 5. Prove that it can take up to 5 guesses (the same player naming coordinates 5 times) before there is a first hit, regardless of what algorithm is used. Give an algorithm for guessing coordinates for which there will always be a hit within the first 5 guesses.

   (b) Suppose now that there are two ships, one of length 5 and one of length 3. Prove that it can still take up to 5 guesses before there is a first hit, regardless of what algorithm is used. Give an algorithm for guessing coordinates for which there will always be a hit within the first 5 guesses.

   (c) Is 5 still the answer if there is a ship of length 5 and another of length 4? Prove your answer.

   (d) Is 5 still the answer if there are two ships of length 5? Prove your answer.

2. Consider algorithms for partitioning a list of $3n$ elements into three sets, each of size n, such that the smallest n elements are all in the first set, the largest n elements in the third set, and the other n elements in the second set. Suppose the algorithm can be modelled by decision trees which have as their basic operations a comparison operation which can compare either 2 and 3 items in one operation and give the ordering between them (the operation will tell which is largest and which is smallest, and this tells which is middle when there are three items).

   (a) Use an information theoretic argument to prove a lower bound on the number of these comparisons any such algorithm would need to make. How many leaves does your tree have and what is the degree of each node in the tree? (You will need to approximate – give a lower bound.)

   (b) Give a linear time algorithm for solving this problem.