Institut for Matematik og Datalogi — January 31, 2011
Syddansk Universitet — JFB

DM508 – Algorithms and Complexity – 2011
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 3. kvartal 2011.* This is identical to the notes from 2010 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 are scheduled to be in U20 on Mondays, 8:15–10 (weeks 5–11), U26 on Thursdays, 10:15–12 (weeks 6, 8, and 10), and U49 on Tuesday, 8:15–10 (week 5), but two of these will be cancelled. The discussion sections will be on Wednesdays, 12:15–14 (weeks 5–11) in U49, Thursdays, 10:15–12 (weeks 7, 9 and 11), in U26, and Friday, 12:15–13 (week 11) in U27a, but two of these will also be cancelled. The "instruktorer" for the course are Magnus Find, Sushmita Gupta, and Abyay Maiti. You may ask any of them (or me) for help, but try to find the "instruktor" who taught the subject you are asking about.

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. 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. To do this, login to Blackboard and find DM508. Note that in the upper left hand corner of the screen, there is an icon which you can click on to expand the the menu for the course. It is just to the left of the code for and name of the course. The assignment hand-in is under "Tools". 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 Magnus, Sushmita, Abyay, 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 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 13:30–14:15 on Tuesdays and 10:30–11:15 on Wednesdays.

There will be an oral exam on April 1 and 2, 2011. 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, January 31

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, February 1

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. We may also begin on NP-completeness, from chapter 34 in the textbook and the section by Papadimitriou and Steiglitz from the course notes.

Problems to be discussed on February 2

Do problems:

1. Do problems 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, rather than to the lower bound mentioned) and 3.10 from pages 140 and 141 of the notes.

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

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 nth 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 Monday, February 14, 8:00

Note that this is part of your exam project, so it must be approved in order for you to take the exam in April, and you may not work with or get help from others not in your group (though you may talk with an "instruktor" 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 as one PDF file.

Suppose you are visiting a city, Fictional, and are currently in a hotel at the corner of 1st Street and 1st Avenue. The roads 1st Street, 2nd Street, 3rd Street, etc., run in the North-South direction and are consecutive, in the logical order. Similarly, the roads 1st Avenue, 2nd Avenue, 3rd Avenue, etc., run in the East-West direction and are consecutive. You discover that something has happened causing the police to block off many roads. All roads north of 1st Avenue, west of 1st Street, east of nth Street, and south of nth Avenue are blocked. In addition, it is impossibly to travel north or west on any street. You want to travel to a museum on the corner of nth Street and nth Avenue. Fortunately, the authorities have set up a Web site where you can check if the ith Street (Avenue) is open between jth Avenue (Street) and $j+1$st Avenue (Street) with a query (i, S, j) (or (i, A, j), respectively). You want to find out if you can get to the museum, and if so, along which streets, using as few queries as possible.

You will probably find this easier if you formulate this problem as a directed graph problem.

1. For $n = 2$, using an adversary argument, show that any algorithm has to make 4 queries in the worst case.

2. For $n > 2$, give an adversary argument showing that any algorithm must do at least $4(n-1)$ queries.

3. For $n > 2$, find an algorithm which uses strictly less than $2n(n-1)$ queries. (Note that this is the maximum possibly number of queries if you check everything. It will probably be helpful to think about what an adversary could do. You might consider checking some edges on the outside boundary first.) It is OK if it is just one less.

4. Prove an information theoretic lower bound for this problem of at least n. You may want to use the inequality $(n/k)^k \leq \binom{n}{k}$.