Institut for Matematik og Datalogi
Syddansk Universitet

October 12, 2011
JFB

Introduction to Computer Science
E11 – Lecture 12

Lecture, October 11, 8:15–10, U26

Rolf Fagerberg finished lecturing on merging and hashing from section 9.5 in the textbook.

Lecture, November 8, 8:15–10, U26

We will begin on chapter 12 in the textbook.

Lecture, November 10, 12:15–14, U26

We will continue with chapter 12 in the textbook, concentrating on security (also from sections 3.5 and 4.5) and cryptography.

Discussion section (and study group before that): November 16 - note location change

All groups should do problem 15 on page 554, so that you can compare solutions.

For the problems below Group 1 should prepare to present Problem 3 on page 530, Problem 6 on page 554. Group 2 should prepare to present Problem 5 on page 530, Problem 9 on page 554, and problem 5 below. Group 3 should prepare to present Problem 3 on page 535, Problems 2 and 14 on pages 553–554.

1. Questions 3 and 5 on page 530.
2. Compare your solutions to problem 15 from page 554.
3. Question 3 on page 535.

4. Do problems 2, 6, 9, and 14 on pages 553–554.
5. Design a Turing machine that once started will use at most three cells on its tape, but will never reach a halt state.
6. Discuss the social issues 1 and 5 on pages 557.

Assignment due 8:15, November 15

Late assignments will not be accepted. Working together is not allowed. (You may write this either in English or Danish.) Explain your answers.

Write your solution to this assignment in LaTeX. Submit a PDF file through the Blackboard system and turn in an identical paper copy. Write your full name and your section number clearly on the first page of your assignment.

1. Assume sets of numbers are represented by lists sorted on element value. For example, the set $\{14, 27, 13, 9, 32\}$ is represented by a list of length 5 containing $[9, 13, 14, 27, 32]$. Write a procedure in pseudocode for constructing $A \cap (B \cap C)$. Use an algorithm similar to that in Figure 9.15 (which goes through each list only once). Assume that you can check for the end of a list, similarly to how you can check for the end of a file, say with "EOL". As in problem 6 (from the discussion section described on the note for Lecture 11), process the three lists simultaneously (you should not first calculate $B \cap C$ and then intersect with A). Make sure your algorithm works correctly in all cases, and explain why it does.
2. If a hash file is partitioned into 12 buckets, what is the probability of at least two of three arbitrary records hashing to the same bucket? Assume that the hash function is such that a randomly chosen record is equally likely to hash to any of the buckets. How many records must be stored in the file until it is more likely for collisions to occur than not? Assume again that there are 12 buckets. (You may use a program to check your answer, but show intermediate steps of your calculations here.)