Institut for Matematik og Datalogi
Syddansk Universitet

October 4, 2010
JFB

Introduction to Computer Science
E10 – Lecture 10

Lecture, October 4, 8:15–10, U37

We finished most of chapter 5 in the textbook (but temporarily skipped the subsection on "Sortware Verification" in section 5.6) and introduced Maple as an example of a programming language (note that there are notes on the course's homepage about Maple). Types and procedures were discussed through examples in Maple.

Lecture, October 7, 14:15–16, U71

Rolf Fagerberg will lecture on section 9.5 in the textbook (merging and hashing).

Lecture, October 11, 8:15–10, U37

We will cover the subsection on "Sortware Verification" in section 5.6. Then, we will begin on chapger 12 in the textbook.

Discussion section: October 13, 12:15–14, U28

The first exercise involves programming. It should be done before you come to discussion section. (You may use Java or Maple or some other language.)

Arun will review the problems from the 4th assignment.

Discuss the following problems from the textbook in groups of three or four.

1. Hashing: Write a program to compute the probability of at least one collision when hashing is used with m records and n buckets. (See the calculation on page 468 of your textbook and generalize it.) Assume

that the the hash function spreads data out essentially randomly. Use your program to answer problem 7 on page 469 and problem 57 on page 478. How did you use your program?

2. Hashing: Question 6 on page 469.

3. Sequential files: Question 3 on page 469 and Problem 54 on page 478.

4. Merging: Question 1 on page 469.

5. Assume sets of numbers are represented by sequential files sorted on element value. For example, the set $\{4, 7, 13, 9, 2\}$ is represented by a sequential file containing $< 2, 4, 7, 9, 13 >$.

   Describe algorithms for constructing $A \cup B$ and $(A \cup B) \cup C$ from A, B and C Note that $(A \cup B) \cup C$ can be done by first computing $A \cup B$ and computing the union of this with C. Instead of giving this solution, process the three files simultaneously, as you do with two files.

6. Assume the database relations A and B each are stored as sequential files of tuples, ordered according to attribute X (which is an attribute of both relations).

   Sketch (details not necessary) an algorithm based on merging for executing the statement

   $$C \leftarrow \text{JOIN } A \text{ and } B \text{ where } A.X = B.X$$

7. Assume again that the database relations A and B each are stored as sequential files, but now no longer ordered on the X attribute.

   Describe an algorithm based on nested loops for executing the statement

   $$C \leftarrow \text{JOIN } A \text{ and } B \text{ where } A.X = B.X$$

   How many comparisons between tuples are performed (as a function of $|A|$ and $|B|$, the numbers of tuples in each relations)?

   Describe how to speed up the algorithm by first using hashing on each relation.

8. Explain how a poorly chosen hash function can result in a hash storage system becoming little more than a sequential file.

9. Discuss questions 3, 6, 7, and 9 on page 479–480.

10. Question 2 on page 469 (it has been mentioned in a previous lecture and is in the sorting simulator you used).