Institut for Matematik og Datalogi October 22, 2014
Syddansk Universitet JFB

Introduction to Computer Science
E14 – Discussion Sections – Week 44

The first exercise involves programming. It should be done before you come to discussion section, possibly in your study group. (You may use Python or Java.)

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 427 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 428 and problem 57 on page 436. How did you use your program?

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

3. Sequential files: Question 3 on page 427 and Problem 54 on page 436.

4. Merging: Question 1 on page 427.

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 the sequence $2, 4, 7, 9, 13$.

   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.