Institut for Matematik og Datalogi
Syddansk Universitet

November 12, 2008
JFB

Introduction to Computer Science
E08 – Week 10

Lecture, November 10

We finished with NP-hardness. We also covered security from the last sections on operating systems and networks and began to introduce cryptography. There were notes on cryptology on the eighth weekly note. In addition, there are links for information on cryptology and PGP on the course's homepage. Rolf Fagerberg lectured on hashing immediately after this lecture.

Lecture, November 17

Daniel Merkle will lecture on bioinformatics.

Lecture, November 19

We will finish with cryptography (from section 12.6 in the textbook and the notes and links mentioned above) and begin on chapter 7 in the textbook.

Lecture, November 24

We will continue with chapter 7 and begin on chapter 8.

Lecture, November 26

Lene Monrad Favrholdt will lecture on on-line algorithms.

Discussion section: first in week 48, in Terminal Room

Read about the class BigInteger in Java's math package before coming to discussion section (you can find it on the Web). Write a function in Java (or Maple) which creates a random integer with k bits and hashes it to an integer in the range from zero to m, inclusive.

Meet in the Terminal Room. Discuss the following problems in groups of two or three.

1. Design three hash functions and program them in Java (or Maple).
2. Test your hash functions. Create an array of length $m + 1$ which will count how many random large integers are hashed to each of the distinct values between zero and m. Look at the results for some relatively small values of m (for example, $m = 4, 8, 12, 20, 50, 100$)
3. With at least four different values of m and some different length large strings (integers) which are hashed, check how many values you need to hash before you get a collision. Does it behave as one would predict with the Birthday Paradox?
4. With at least four different values of m and some different length large strings, check how even your distribution is? What is the difference between the largest number of strings hashing to any particular value and the smallest number? How does this seem to depend on the number of strings you hash?
5. With your hash functions, how would you find collisions (two strings that hash to the same value) when m is very large (say 2^{80})? Are your hash functions cryptographically secure?

Discussion section: second in week 48, in Terminal Room

Meet in the Terminal Room. Arun Vadiveal will give an introduction to a version control system, Subversion (SVN), which you will then try using. He has written notes which are on the course homepage.

Remember the assignment due November 24.