

## Homework 6 – Due Friday, March 9, 2007

### New!

- To facilitate grading, please write down your solution to each problem on a separate sheet of paper. Make sure to include all identifying information and your collaborators on each sheet. Your solutions to different problems will be graded separately, possibly by different people, and returned to you independently of each other.
- When you are done with Problem 2, please enter your answers to this problem on Angel. The instructions will be posted on Angel. We will try using Angel to grade parts of this problem, but if for some reason it is not successful, we will use your write up.

**Reading** CLRS Chapters 11.1 – 11.3 (excluding 11.3.3), 11.4 (only pages 237-238), 12.1 – 12.3

**Exercises** These should not be handed in, but the material they cover may appear on exams.

- Hashing: CLRS Ex. 11.1-1, 11.1-4, 11.2-2, 11.2-3, 11.2-5, 11.3-1, 11.3-3, 11.3-4, 11.4-1.
- Binary Search Trees: CLRS Problem 2-4.

### Problems to be handed in

1. **(Direct Addressing)** CLRS 11.1-3. Write pseudocode for initializing your data structure and all three dictionary operations.
2. **(Birthday Problem)** Based on CLRS 11.2-1. Suppose we use a hash function  $h$  to hash  $n$  distinct keys into an array of length  $m$ . Assuming uniform hashing, let  $X$  be the number of collisions, that is, the number of pairs of keys  $\{k, \ell\}$  with  $k \neq \ell$  and  $h(k) = h(\ell)$ .
  - (a) For  $1 \leq i < j \leq n$ , let  $X_{ij}$  be the indicator variable for the event that the  $i$ -th and the  $j$ -th key hash to the same value. What is the expectation of  $X_{ij}$ ?
  - (b) Express  $X$  in terms of  $X_{ij}$ 's and compute the expectation of  $X$  as a function of  $m$  and  $n$ . (Hint: use linearity of expectation.)
  - (c) Suppose you want to store records for all students in CSE465 in a hash table. You choose the hash function to return the student's birthday (only month and day, no year). What does simple uniform hashing assumption mean for this hash function? Roughly how many student records will cause your hash function to have 1 collision in expectation? How many collisions do you expect to see in our class (70 students)?
3. **(Choosing a Hash Function)** Read CLRS Chapter 11.3, pages 229–232.

- (a) CLRS 11.3-2
  - (b) CLRS 11.3-3
4. (**Binary Search Trees**) CLRS 12.1-3 and 12.1-4.

## Programming Assignment 1: Anagrams Revisited

The first part of the assignment does not involve any programming. It should be turned in together with homework 6. It will not be graded: you will get full credit just for turning in meaningful answers for all the questions.

The second part is the programming assignment. It is due **Wednesday, March 21, 2007 on Angel**. However, you are strongly encouraged to make sure that by Friday, March 9 you are in the coding phase and you have no technical issues that might prevent you from working on the assignment.

Our usual collaboration and honesty policy applies to this assignment. In particular, you can work on it with at most 3 other students currently enrolled in CSE465. You can discuss an aspect of this assignment only after thinking about it by yourself for at least 45 minutes. You must clearly identify your collaborators. The programs should be written individually. Copying code from the web or other students in the class is strictly prohibited.

1. Given two files with English text, you would like to determine if one can be obtained from the other by changing the word order. Give an algorithm for the problem, using hashing with chaining. Your algorithm should take expected  $O(n)$  time (under the simple uniform hashing assumption), where  $n$  is the number of words in both files. Assume that each word is at most 32 characters long. What hash function do you propose to use? Give pseudocode for your algorithm.
2. Please write a program in C++ or Java implementing the algorithm above. Your program should take two files called *doc1* and *doc2* as input and return true or false. You will be provided with test files on Angel. If you choose to program in Java, we strongly encourage you to look at Java implementations of some algorithms we covered (see the CD that came with CLRS). You may assume that the input contains only lower case letters and spaces.