

CSE 326 DATA STRUCTURES HOMEWORK 5

Due: **Friday, February 16, 2007** at the beginning of class in the lecture section you are registered in
Please put your quiz section (AA,AB,BA,BB) in addition to your name at the top of your homework

Problem 1. Universal Hashing

Hash functions usually give us very good performance. But one weakness of the fixed hash functions we have discussed so far is that a malicious set of inputs can lead to $O(n)$ behavior. If someone is trying to slow down your system, and they have access to submitting inputs to a hash function, they can design a set of inputs which causes $O(n)$ behavior (even if you spot duplicate inputs.) Is there any way to avoid this?

In this problem we will discuss a method for overcoming this problem called *universal hashing*. Universal hashing is a randomized algorithm for selecting a hash function with the following property: for any two distinct inputs, the probability that there is a hash collision between these inputs is the same as if the function was a random function. Thus, if h has function values in a range of size r , the probability of any particular hash collision should be $\frac{1}{r}$.

Let p be a prime number and suppose that we are hashing a pair of numbers x and y which are each between 0 and $p - 1$ to a range of size p . In other words our hash function has a domain made up of pairs of numbers x and y between 0 and $p - 1$ and a range of numbers between 0 and $p - 1$. To be clear the domain is of size p^2 and the range of size p . For our hashing function we randomly (and independently) select two numbers a and b , which are both between 0 and $p - 1$ and then use these to construct the hash function (this is the universal hashing step):

$$h(x, y) = (ax + by) \bmod p.$$

- (a) Fix two non-identical inputs (x_1, y_1) and (x_2, y_2) . (They may have the same x or y values, but not both.) For how many distinct pairs of our random numbers, a and b will these two inputs hash to the same slot?
- (b) If I choose each of a and b uniformly at random from $0, 1, \dots, p - 1$ what is the probability that the two non-identical (x_1, y_1) and (x_2, y_2) will hash to the same value?
- (c) Given an arbitrary set of n distinct inputs (x_i, y_i) , what is the expected number (over my random choices of a and b each uniformly from 0 to $p - 1$) of pairs i, j , $i < j$, for which $h(x_i, y_i) = h(x_j, y_j)$? (Recall that the expectation value of a set of random variables X which occur with probability p_X is $\langle X \rangle = \sum_X p_X X$.)

Problem 2. A Future Hash Table Problem

Later in life you are charged with writing a hash table which uses separate chaining in a very obscure language which you are not familiar with. Since you're not familiar with the language you can't figure out how to get your linked list working properly when dealing with collisions. So you decide that maybe by making your hash table big enough you will make it such that the chance of a collision will be negligible and thus your deficiency won't be spotted! You know that the program you are writing will be given k elements to hash so you decide that you'll use a hash table containing k^2 locations. For parts a,b, and c below express your answer as a function of k in as simple terms (most reduced) as you can.

- (a) How many different ways can the k different elements be hashed into the k^2 different table locations?

- (b) How many of these ways result in no collisions?
- (c) If each way has the same probability, then what is the probability of one or more collisions?
- (d) Compute this probability for $k = 5$.

Problem 3. Unions

Do Weiss Problem 8.1.