Solution to Homework 5

Problem 1 (CLRS 11.2-1). (1 point) Suppose we use a hash function $h$ to hash $n$ distinct keys into an array $T$ of length $m$. Assuming simple uniform hashing, what is the expected number of collisions? More precisely, what is the expected cardinality of $\{\{k, l\} : k \neq l$ and $h(k) = h(l)\}$?

Solution: Let us define $X_{ij}$ as a random variable indicating $i$-th and $j$-th keys (by insertion order) being hashed to the same location:

$$X_{ij} = I\{h(k_i) = h(k_j)\}.$$ 

Then the number of collisions $N_c$ can be expressed as sum of $X_{ij}$ over all pairs of distinct keys:

$$N_c = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}.$$ 

Using linearity of expectation and simple uniform hashing:

$$E[N_c] = E\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{m} = \frac{n(n-1)}{2m}.$$ 

Alternatively:

$$E[N_c] = \binom{n}{2} \frac{1}{m} = \frac{n(n-1)}{2} \frac{1}{m} = \frac{n(n-1)}{2m}.$$ 

Problem 2 (CLRS 11.3-3). (3 points) Consider a version of the division method in which $h(k) = k \mod m$, where $m = 2^p - 1$ and $k$ is a character string interpreted in radix $2^p$. Show that if we can derive string $x$ from string $y$ by permuting its characters, then $x$ and $y$ hash to the same value. Give an example of an application in which this property would be undesirable in a hash function.

Solution: Any permutation of a string can be obtained by repeated exchanges of pairs of characters. Thus, it suffices to show that strings $x$ and $y$ derived from $x$ by exchanging a single pair of characters hash to the same value.
Let us define \( x \) and \( y \) as identical strings of \( n \) characters with a single pair of characters inter-changed:

\[
x_a = y_b, \\
y_a = x_b. 
\]

\( x \) and \( y \) have the following representations in radix \( 2^p \):

\[
x = \sum_{i=0}^{n-1} x_i 2^{ip}, \\
y = \sum_{i=0}^{n-1} y_i 2^{ip}.
\]

And the following hash values:

\[
h(x) = \left( \sum_{i=0}^{n-1} x_i 2^{ip} \right) \mod (2^p - 1), \\
h(y) = \left( \sum_{i=0}^{n-1} y_i 2^{ip} \right) \mod (2^p - 1).
\]

We know that:

\[
0 \leq h(x) < 2^p - 1, \\
0 \leq h(y) < 2^p - 1, \\
-(2^p - 1) < h(x) - h(y) < 2^p - 1.
\]

To show that \( h(x) = h(y) \) it is therefore sufficient to show that:

\[
(h(x) - h(y)) \mod (2^p - 1) = 0.
\]

The characters in \( x \) and \( y \) are the same except for \( x_a, x_b, y_a, \) and \( y_b \). Thus, the sums in radix \( 2^p \) representation will also be the same (and will cancel out on subtraction) except for \( x_a 2^{ap}, x_b 2^{bp}, y_a 2^{ap}, \) and \( y_b 2^{bp} \). We also recall that \( x_a = y_b \) and \( y_a = x_b \), and obtain:

\[
(h(x) - h(y)) \mod (2^p - 1) = \left( (x_a 2^{ap} + x_b 2^{bp}) - (y_a 2^{ap} + y_b 2^{bp}) \right) \mod (2^p - 1) \\
= \left( (x_a 2^{ap} + x_b 2^{bp}) - (x_b 2^{ap} + x_a 2^{bp}) \right) \mod (2^p - 1) \\
= \left( (x_a - x_b) 2^{ap} - (x_a - x_b) 2^{bp} \right) \mod (2^p - 1) \\
= \left( (x_a - x_b)(2^{ap} - 2^{bp}) \right) \mod (2^p - 1) \\
= \left( (x_a - x_b) 2^{bp}(2^{(a-b)p} - 1) \right) \mod (2^p - 1).
\]
By the sum of geometric series:

\[ \sum_{i=0}^{a-b-1} 2^i = \frac{2^{(a-b)p} - 1}{2^p - 1}, \]

\[ 2^{(a-b)p} = \left( \sum_{i=0}^{a-b-1} 2^i \right) (2^p - 1). \]

And we can rewrite our expression for \( h(x) - h(y) \) as:

\[
\left( h(x) - h(y) \right) \mod (2^p - 1) = \left( (x_a - x_b)2^{bp} \left( \sum_{i=0}^{a-b-1} 2^i \right) (2^p - 1) \right) \mod (2^p - 1).
\]

One of the factors in \( h(x) - h(y) \) is \( 2^p - 1 \), so:

\[
\left( h(x) - h(y) \right) \mod (2^p - 1) = 0,
\]

and therefore \( x \) and \( y \) hash to the same value.

This property would be highly undesirable in any application hashing strings that are likely to be permutations of each other. Consider bit strings of length 32. Among \( 2^{32} \) possible strings there are only 33 distinct sets of bits. For example, all \( \binom{32}{16} = 601080390 \) strings with 16 bits on and 16 bits off will hash to the same value.

**Problem 3 (CLRS 11.4-2).** (2 points) Write pseudocode for Hash-Delete as outlined in the text, and modify Hash-Insert to handle the special value Deleted.

**Solution:** To delete, we locate the element using hash-search (CLRS p. 271), and mark it with a special value DELETED:

```python
def hash-delete(T, k):
    j = hash-search(T, k)
    if j == NULL:
        return
    T[j] = DELETED
```

The only modification to hash-insert (CLRS p. 270) is to treat slots containing deleted elements as empty, so they can be reused:

```python
def hash-insert(T, k):
    i = 0
    repeat:
        j = h(k, i)
        if T[j] == NULL or T[j] == DELETED:
            T[j] = k
            return j
        else:
```

3
Problem 4 (CLRS 11.2). (5 points) Suppose that we have a hash table with \( n \) slots, with collisions resolved by chaining, and suppose that \( n \) keys are inserted into the table. Each key is equally likely to be hashed to each slot. Let \( M \) be the maximum number of keys in any slot after all the keys have been inserted. Your mission is to prove an \( O(\log n / \log \log n) \) upper bound on \( E[M] \), the expected value of \( M \).

Solution:

(a) Argue that the probability \( Q_k \) that exactly \( k \) keys hash to a particular slot is given by

\[
Q_k = \left( \frac{1}{n} \right)^k \left( 1 - \frac{1}{n} \right)^{n-k} \binom{n}{k}.
\]

Given that each key is equally likely to be hashed to each of \( n \) slots:

\[
\Pr \{k \text{ keys hashed to the same slot}\} = \left( \frac{1}{n} \right)^k,
\]

\[
\Pr \{\text{other } n-k \text{ keys hashed to other slots}\} = \left( 1 - \frac{1}{n} \right)^{n-k}.
\]

Observing that there are \( \binom{n}{k} \) ways to pick \( k \) out of \( n \) keys gives:

\[
Q_k = \left( \frac{1}{n} \right)^k \left( 1 - \frac{1}{n} \right)^{n-k} \binom{n}{k}.
\]

(b) Let \( P_k \) be the probability that \( M = k \), that is, the probability that the slot containing the most keys contains \( k \) keys. Show that \( P_k \leq nQ_k \).

Let us define \( X_i \) as a random variable denoting the number of keys in slot \( i \). Then:

\[
M = \max_{1 \leq i \leq n} \{X_i\},
\]

\[
P_k = \Pr \{M = k\}
\]

\[
= \Pr \left\{ \max_{1 \leq i \leq n} (X_i) = k \right\}.
\]

For \( X_i \) to achieve maximum in \( i = j \):

\[
P_k = \Pr \{X_j = k\} \cdot \prod_{i=1}^{n} \Pr \{X_j \leq k\}
\]

\[
\leq \Pr \{X_j = k\}
\]

\[
\leq \sum_{i=1}^{n} \Pr \{X_i = k\}.
\]
By (a), the probability that exactly \( k \) keys hash to a particular slot is \( Q_k \). Thus:

\[
P_k \leq \sum_{i=1}^{n} \Pr\{X_i = k\}
\]

\[
= \sum_{i=1}^{n} Q_k
\]

\[
= nQ_k.
\]

(c) Use Stirling’s approximation, equation (3.18) in CLRS, to show that \( Q_k < e^k/k^k \).

Observing that \( \left(1 - \frac{1}{n}\right)^{n-k} < 1 \), we have:

\[
Q_k = \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \binom{n}{k}
\]

\[
< \left(\frac{1}{n}\right)^k \binom{n}{k}.
\]

We now note that:

\[
\binom{n}{k} = \frac{n!}{k!(n-k)!}
\]

\[
= \frac{1}{k!} \frac{n(n-1)...(n-k+1)(n-k)...1}{(n-k)!}
\]

\[
= \frac{1}{k!} \left(n(n-1)...(n-k+1)\right)
\]

\[
< \frac{1}{k!} n^k.
\]

And thus:

\[
Q_k < \left(\frac{1}{n}\right)^k \frac{1}{k!} n^k = \frac{1}{k!}.
\]

Finally, we use Stirling’s approximation:

\[
n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right),
\]

to note that:

\[
n! > \left(\frac{n}{e}\right)^n,
\]

and therefore:

\[
Q_k < \frac{1}{k!} < \frac{1}{\left(\frac{k}{e}\right)^k} = \frac{e^k}{k^k}.
\]
(d) Show that there exists a constant $c > 1$ such that $Q_{k_0} < 1/n^3$ for $k_0 = c \lg n / \lg \lg n$. Conclude that $P_k < 1/n^2$ for $k \geq k_0 = c \lg n / \lg \lg n$.

By (c):

$$Q_{k_0} < \frac{e^{k_0}}{k_0^{k_0}},$$

so to show $Q_{k_0} < 1/n^3$ we can show:

$$\frac{e^{k_0}}{k_0^{k_0}} \leq \frac{1}{n^3},$$

$$n^3 \leq \frac{k_0^{k_0}}{e^{k_0}}.$$

Taking logarithms, using definition of $k_0$, and simplifying:

$$3 \lg n \leq \lg k_0 k_0 - \lg e^{k_0}$$

$$= k_0 (\lg k_0 - \lg e)$$

$$= \frac{c \lg n}{\lg \lg n} \left( \lg \left( \frac{c \lg n}{\lg \lg n} \right) - \lg e \right)$$

$$= \frac{c \lg n}{\lg \lg n} \left( \lg c \lg n - \lg (\lg \lg n) - \lg e \right)$$

$$= \frac{c \lg n}{\lg \lg n} \left( \lg c + \lg \lg n - \lg \lg \lg n - \lg e \right),$$

$$3 \leq \frac{c}{\lg \lg n} \left( \lg c + \lg \lg n - \lg \lg \lg n - \lg e \right)$$

$$= c \left( 1 + \frac{\lg c - \lg e}{\lg \lg n} - \frac{\lg \lg \lg n}{\lg \lg n} \right).$$

We note that the right-hand side is only defined for $n > 2$, and that as $n \to \infty$, the logarithm ratios go to zero, sending the parenthesized expression to 1. Defining $n_0 > 2$ such that:

$$1 + \frac{\lg c - \lg e}{\lg \lg n} - \frac{\lg \lg \lg n}{\lg \lg n} \geq \frac{1}{2} \quad \text{for all } n \geq n_0,$$

we can see that:

$$3 \leq c \left( 1 + \frac{\lg c - \lg e}{\lg \lg n} - \frac{\lg \lg \lg n}{\lg \lg n} \right) \quad \text{for any } c \geq 6 \text{ and all } n \geq n_0.$$

For $n$ smaller than $n_0$, we note that the inequality:

$$3 \leq c \left( 1 + \frac{\lg c - \lg e}{\lg \lg n} - \frac{\lg \lg \lg n}{\lg \lg n} \right) \quad \text{for } 2 < n < n_0.$$
has \( n_0 - 3 \) solutions in \( c \), and by defining \( c_{\text{max}} \) to be the largest \( c \) out of \( n_0 - 3 \) solutions, we can pick the largest of 6 and \( c_{\text{max}} \) to satisfy the inequality on the entire range \( n > 2 \).

Having found a constant \( c \) such that \( Q_{k_0} < 1/n^3 \) for \( k_0 = c \lg n / \lg \lg n \), we now need to show that \( P_k < 1/n^2 \) for \( k \geq k_0 \).

By (b), \( P_k \leq nQ_k \) and so \( P_{k_0} \leq nQ_{k_0} \). By first part of (d), \( Q_{k_0} < 1/n^3 \) and so:

\[
P_{k_0} \leq nQ_{k_0} < n \frac{1}{n^3},
\]
\[
P_{k_0} < \frac{1}{n^2}.
\]

This shows that the inequality holds for \( k = k_0 \). We will now show that it holds for \( k > k_0 \) by showing that \( Q_k < 1/n^3 \) for all \( k \geq k_0 \).

By picking \( c \) inside \( k_0 \) large enough that \( k_0 > e \), we have \((e/k) < 1\) and \((e/k)^{m+1} < (e/k)^m\) for \( k \geq k_0 \) and any \( m \). Using \( Q_k < e^k/k^k \) from (c):

\[
Q_k < \frac{e^k}{k^k} \leq \frac{e^{k_0}}{k_0},
\]
\[
Q_k < \frac{e^{k_0}}{k_0}.
\]

Combining with \( Q_{k_0} < 1/n^3 \) from the first part of (d) and keeping in mind that \( k \geq k_0 \):

\[
Q_k < \frac{e^{k_0}}{k_0} \quad \text{and} \quad Q_{k_0} < 1/n^3,
\]
\[
Q_k < \frac{1}{n^3}.
\]

(e) Argue that

\[
E[M] \leq \Pr \left\{ M > \frac{c \lg n}{\lg \lg n} \right\} \cdot n + \Pr \left\{ M \leq \frac{c \lg n}{\lg \lg n} \right\} \cdot \frac{c \lg n}{\lg \lg n}.
\]

Conclude that \( E[M] = O(\lg n / \lg \lg n) \).

By definition of \( M \) and definition of expectation:

\[
E[M] = \sum_{k=0}^{n} k \cdot \Pr \{ M = k \}.
\]

Or, splitting the sum on \( k_0 \):

\[
E[M] = \sum_{k=0}^{k_0} k \cdot \Pr \{ M = k \} + \sum_{k=k_0+1}^{n} k \cdot \Pr \{ M = k \}.
\]
The number of keys in a slot cannot exceed the total number of keys so far, thus:

\[ E[M] \leq \sum_{k=0}^{k_0} k_0 \cdot \Pr\{M = k\} + \sum_{k=k_0+1}^{n} n \cdot \Pr\{M = k\}. \]

Simplifying and expanding \( k_0 \):

\[
E[M] \leq k_0 \sum_{k=0}^{k_0} \Pr\{M = k\} + n \sum_{k=k_0+1}^{n} \Pr\{M = k\}
\]

\[
= k_0 \cdot \Pr\{M \leq k_0\} + n \cdot \Pr\{M > k_0\}
\]

\[
= k_0 \cdot \Pr\{M \leq k_0\} + n \cdot \Pr\{M > k_0\}
\]

\[
= \frac{c \lg n}{\lg \lg n} \cdot \Pr\left\{ M \leq \frac{c \lg n}{\lg \lg n} \right\} + n \cdot \Pr\left\{ M > \frac{c \lg n}{\lg \lg n} \right\}. \]

To show that \( E[M] = O(\lg n/\lg \lg n) \), we first rewrite \( \Pr\{M > k_0\} \) in terms of \( P_k \) and apply \( P_k < 1/n^2 \) from (d):

\[
\Pr\{M > k_0\} = \sum_{k=k_0+1}^{n} \Pr\{M = k\}
\]

\[
= \sum_{k=k_0+1}^{n} P_k
\]

\[
< \sum_{k=k_0+1}^{n} \frac{1}{n^2}
\]

\[
< \frac{1}{n^2}
\]

\[
= \frac{1}{n}.
\]

Finally, using a trivial upper bound \( \Pr\{M \leq k_0\} \leq 1 \):

\[
E[M] \leq k_0 \cdot \Pr\{M \leq k_0\} + n \cdot \Pr\{M > k_0\}
\]

\[
= k_0 \cdot 1 + n \cdot \frac{1}{n}
\]

\[
= k_0 + 1
\]

\[
= \frac{c \lg n}{\lg \lg n} + 1
\]

\[
= O\left( \frac{\lg n}{\lg \lg n} \right).
\]