Assignment 5

Benjamin Jakubowski

June 30, 2016

1. Expected number of collisions

Consider a hash function h to hash n distinct keys into an array T of length m. Assuming simple uniform hashing, the expected number of collisions is $\frac{n^2-n}{2m}$.
To see this, note the number of collisions is the cardinality of

$$C = \{\{k, l\} : k \neq l \text{ and } h(k) = h(l)\}$$

Then let

$$X_{kl} = \mathbb{1}(h(k) = h(l))$$

Then

$$\begin{aligned}
E[|C|] &= E\left[\sum_{k=1}^{n}\sum_{l=k+1}^{n} X_{kl}\right] \\
&= \sum_{k=1}^{n}\sum_{l=k+1}^{n} E[X_{kl}] \qquad \text{by linearity of expectation} \\
&= \sum_{k=1}^{n}\sum_{l=k+1}^{n} \frac{1}{m} \qquad \text{under assumption of simple uniform hashing} \\
&= \frac{1}{m}\sum_{k=1}^{n}(n-k) \\
&= \frac{1}{m}\left(n^2 - \sum_{k=1}^{n} k\right) \\
&= \frac{1}{m}\left(n^2 - \frac{n(n+1)}{2}\right) \\
&= \frac{n^2-n}{2m}
\end{aligned}$$

2. Permutations of strings collide

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 in radix 2^p. We show that if we can derive string y from string x (**note I have switched x and y relative to the question**) by permuting the characters in x, then x and y hash to the same value.
First, let

$$x = x_{n-1}x_{n-2}\cdots x_1x_0$$

In radix 2^p, we interpret this string as

$$\sum_{i=0}^{n-1} x_i(2^p)^i = \sum_{i=0}^{n-1} x_i 2^{ip}$$

Note we can obtain any permutation of x by sequentially swapping adjacent characters[1]. Thus, now letting y be a permutation obtained from an arbitrary string x by swapping two adjacent characters, if we show y and x hash to the same value, we have shown that all permutations of our original string x hash to the same value.
Thus, we proceed by letting

$$x = x_{n-1}\cdots x_{u+1}x_u\cdots x_1x_0$$

$$y = x_{n-1}\cdots x_u x_{u+1}\cdots x_1x_0$$

Then (in radix 2^p) y is interpreted as

$$\left(\sum_{i=0, i\neq u, u+1}^{n-1} x_i 2^{ip}\right) + x_u 2^{(u+1)p} + x_{u+1}2^{up}$$

Now we need to show $h(x) = h(y)$, or equivalently $h(x) - h(y) = 0$.
Well,

[1]See the Steinhaus-Johnson-Trotter algorithm, `https://en.wikipedia.org/wiki/Steinhaus%E2%80%93Johnson%E2%80%93Trotter_algorithm`

$$
\begin{aligned}
h(x) - h(y) &= \left[\sum_{i=0}^{n-1} x_i 2^{ip}\right] \mod (2^p - 1) \\
&\quad - \left[\left(\sum_{\substack{i=0 \\ i \neq u, u+1}}^{n-1} x_i 2^{ip}\right) + x_u 2^{(u+1)p} + x_{u+1} 2^{up}\right] \mod (2^p - 1) \\
&= \left[\sum_{i=0}^{n-1} x_i 2^{ip} - \left(\sum_{\substack{i=0 \\ i \neq u, u+1}}^{n-1} x_i 2^{ip}\right) - x_u 2^{(u+1)p} - x_{u+1} 2^{up}\right] \mod (2^p - 1) \\
&= \left[(x_{u+1} 2^{(u+1)p} + x_u 2^{up}) - (x_u 2^{(u+1)p} + x_{u+1} 2^{up})\right] \mod (2^p - 1) \\
&= (x_{u+1} - x^u)(2^{(u+1)p} - 2^{up}) \mod (2^p - 1) \\
&= (x_{u+1} - x^u) 2^{up}(2^p - 1) \mod (2^p - 1) \\
&= 0
\end{aligned}
$$

Thus, $h(x) - h(y) = 0 \implies h(x) = h(y)$, so these strings hash to the same value.

3. Supporting `DELETE` in an open-addressed hash table

To support `delete` in an open-addressed hash table, we need to amend the `insert` function, and write new (psuedo-)code for the `delete` function.

```
function HASH_DELETE(T, k)
    search_result = hash_search(T, k)
    if search_result == NIL then
        return "k not in T"
    else
        T[search_result] = DELETED
        return "k deleted from T"
function HASH_INSERT(T, k)
    i = 0
    while i < m do
        j = h(k, i) // where h is our hash function
        if T[j] == NIL or T[j] == DELETED then
            T[j] = k
            return j
        else
            i ++
    error "hash table overflow"
```

4. Proving an $O(\lg n/\lg\lg n)$ upper bound on $E[M]$

Suppose 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.

a. Probability Q_k that k keys hash to a particular slot

First, note the number of keys K that hash to a particular slot is a binomial random variable. Thus,

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

b. Showing $P_k \le nQ_k$

Now let P_k be the probability that $M = k$, or the probability the slot containing the most keys contains k keys. We aim to show $P_k \le nQ_k$.
First, let K_i be the number of keys hashed to slot i. Next, let A be the event $M = k$. Then note A is the union of the events

$$A = \bigcup_{i=1}^{n} A_i = \bigcup_{i=1}^{n} [K_i = k \text{ and } K_j \le k \text{ for all } j \ne i]$$

Then, by the union bound, we have

$$\begin{aligned}
P(M = k) &= P(A) = P(\cup_{i=1}^{n} A_i) \\
&\le \sum_{i=1}^{n} P(A_i) \qquad \text{by the union bound} \\
&= \sum_{i=1}^{n} P(K_i = k \text{ and } K_j \le k \text{ for all } j \ne i) \\
&= \sum_{i=1}^{n} P(K_i = k)\underbrace{P(K_j \le k \text{ for all } j \ne i \mid K_i = k)}_{\le 1,\text{ since a probability}} \\
&\le \sum_{i=1}^{n} P(K_i = k) \\
&= \sum_{i=1}^{n} Q_k \\
&= nQ_k
\end{aligned}$$

C. Showing $Q_k < e^k/k^k$

Now we show $Q_k < e^k/k^k$. Before proceeding, here's a brief outline of our attack:

- First, we use the inequality $n! < n^n$ to show $Q_k \leq 1/k!$
- Then we apply Stirling's inequality to obtain the desired result

To begin

$$
\begin{aligned}
Q_k &= \binom{n}{k}\left(\frac{1}{n}\right)^k\left(1-\frac{1}{n}\right)^{n-k} \\
&= \left(\frac{1}{n}\right)^k\left(\frac{n-1}{n}\right)^{n-k}\frac{n!}{(n-k)!k!} \\
&= \left(\frac{1}{n}\right)^k\left(\frac{n-1}{n}\right)^{n-k}\left(\prod_{i=(n-k)+1}^{n}\underbrace{i}_{\leq n}\right)\frac{1}{k!} \\
&\leq \left(\frac{1}{n}\right)^k\left(\frac{n-1}{n}\right)^{n-k}\left(\prod_{i=(n-k)+1}^{n} n\right)\frac{1}{k!} \\
&= \left(\frac{1}{n}\right)^k\left(\frac{n-1}{n}\right)^{n-k} n^k\frac{1}{k!} \\
&= \underbrace{\left(\frac{n-1}{n}\right)^{n-k}}_{<1}\frac{1}{k!} \\
&\leq \frac{1}{k!}
\end{aligned}
$$

Next, we apply Sterling's approximation- recall

$$k! = \sqrt{2\pi k}\left(\frac{k}{e}\right)^k\left(1+\Theta\left(\frac{1}{k}\right)\right)$$

Thus,

$$
\begin{aligned}
\frac{1}{k!} &= \underbrace{\frac{1}{\sqrt{2\pi k}\left(1+\Theta\left(\frac{1}{k}\right)\right)}}_{<1}\frac{e^k}{k^k} \\
&< \frac{e^k}{k^k}
\end{aligned}
$$

yielding our final result

$$Q_k < \frac{e^k}{k^k}$$

D. Showing $P_k < 1/n^2$ for $k \geq k_0$

We first show there exists some $c > 1$ such that $Q_{k_0} < 1/n^3$ for $k_0 = c \lg n / \lg\lg n$. From (c), we know $Q_k < \frac{e^k}{k^k}$. Thus, to show $Q_{k_0} < 1/n^3$ we must find c such that

$$
\begin{aligned}
& Q_{k_0} < \frac{e^{k_0}}{{k_0}^{k_0}} < 1/n^3 \\
\Longrightarrow \quad & \lg\left(\frac{e^{k_0}}{{k_0}^{k_0}}\right) < \lg\left(1/n^3\right) \\
\Longrightarrow \quad & k_0 - k_0 \lg(k_0) < \lg(1) - 3\lg(n) \\
\Longrightarrow \quad & k_0(1 - \lg(k_0)) < -3\lg(n) \\
\Longrightarrow \quad & \frac{c \lg n}{\lg\lg n}\left(1 - \lg\left(\frac{c\lg n}{\lg\lg n}\right)\right) < -3\lg(n) \\
\Longrightarrow \quad & \frac{c \lg n}{\lg\lg n}(1 - (\lg(c) + \lg\lg n - \lg\lg\lg n)) < -3\lg(n) \\
\Longrightarrow \quad & \frac{c}{\lg\lg n}(1 - (\lg(c) + \lg\lg n - \lg\lg\lg n)) < -3 \\
\Longrightarrow \quad & c + \frac{c\lg c}{\lg\lg n} - \frac{c}{\lg\lg n} - \frac{c\lg\lg\lg n}{\lg\lg n} > 3
\end{aligned}
$$

Now, note that in the limit (as $n \to \infty$), the left hand side goes to c (as the remaining terms approach zero). Thus, there clearly exists a constant that ensures this inequality holds for large n (say $n > n_0$)- let's call one such constant c'.
Now consider $n \leq n_0$. Note this is a finite set (since it is bounded above by n_0). Thus, if we can find a solution c_i for each n, and then take $c = \max\{c', c_i \text{ for all } i \leq n_0\}$, then we've found a constant that ensures the desired inequality for all n.
Thus, to proceed note

$$
\begin{aligned}
& c + \frac{c\lg c}{\lg\lg n} - \frac{c}{\lg\lg n} - \frac{c\lg\lg\lg n}{\lg\lg n} > 3 \\
\Longrightarrow \quad & c + c\left[\frac{\lg c - 1 - \lg\lg\lg n}{\lg\lg n}\right] > 3
\end{aligned}
$$

This is clearly true if

- $c > 3$
- $\lg\lg n > 0$
- $\lg c > 1 + \lg\lg\lg n$

Since we can find a c that meets these requirements for all $n \leq n_0$ [2], and the set $n \leq n_0$ is finite, our final constant c is simply

$$c = \max\{c', c_i \text{ for all } i \leq n_0\}$$

Finally, we conclude $P_k < 1/n^2$ for $k \geq k_0 = c \lg n / \lg \lg n$. To see this, note we just showed

$$Q_{k_0} < e^{k_0}/{k_0}^{k_0} < 1/n^3$$

Since e^k/k^k decreases monotonically with k, for all $k > k_0$.

$$Q_k < e^k/k^k \leq e^{k_0}/{k_0}^{k_0} = Q_{k_0} < 1/n^3$$

Thus, applying the result from part (b), we have

$$P_k \leq nQ_k \leq nQ_{k_0} < n1/n^3 = 1/n^2$$

E. Bounding $E[M] = O(\lg n / \lg \lg n)$

Now we conclude by placing the desired bound on $E[M]$.
First, note

$$\begin{aligned}
E[M] &= \sum_k k \cdot P(M = k) \\
&= \sum_{k \leq \frac{c \lg n}{\lg \lg n}} \underbrace{k}_{\leq \frac{c \lg n}{\lg \lg n}} P(M = k) + \sum_{k > \frac{c \lg n}{\lg \lg n}}^{n} \underbrace{k}_{< n} P(M = k) \\
&\leq \frac{c \lg n}{\lg \lg n} \cdot \sum_{k \leq \frac{c \lg n}{\lg \lg n}} P(M = k) + n \cdot \sum_{k > \frac{c \lg n}{\lg \lg n}}^{n} P(M = k) \\
&= \frac{c \lg n}{\lg \lg n} \cdot P\left(k \leq \frac{c \lg n}{\lg \lg n}\right) + n \cdot P\left(k > \frac{c \lg n}{\lg \lg n}\right)
\end{aligned}$$

But, from part (d), we know $P_k \leq 1/n^2$ for all $k > k_0$. Thus

$$P\left(k > \frac{c \lg n}{\lg \lg n}\right) = \sum_{k > \frac{c \lg n}{\lg \lg n}}^{n} P(M = k) \leq n * 1/n^2 = 1/n$$

[2] I am aware my argument requires $n > e$. However, this doesn't seem problematic, since our end goal is to put a big-O bound on $E[M]$, so failing to address the cases where $n = 1$ and $n = 2$ seems largely irrelevant.

Thus

$$
\begin{aligned}
E[M] &\leq \frac{c \lg n}{\lg \lg n} \cdot P\left(k \leq \frac{c \lg n}{\lg \lg n}\right) + n \cdot P\left(k > \frac{c \lg n}{\lg \lg n}\right) \\
&\leq \frac{c \lg n}{\lg \lg n} \cdot P\left(k \leq \frac{c \lg n}{\lg \lg n}\right) + n \cdot (1/n) \\
&\leq \frac{c \lg n}{\lg \lg n} \cdot \underbrace{P\left(k \leq \frac{c \lg n}{\lg \lg n}\right)}_{=c_2} + 1 \\
&= (c \cdot c_2) \frac{\lg n}{\lg \lg n} + 1
\end{aligned}
$$

Giving us our desired result:

$$E[M] = O(\lg n / \lg \lg n)$$