13 Open addressing: linear rehashing

(13.1) Definition *When i is the index in the hash table (of size m), by abuse of notation $i+c$ implies addition with wraparound:*

$$i + c \equiv (i + c) \mod m$$

The most challenging scheme to analyse (with the worst performance) is *linear rehashing.* Under this scheme, in searching for a key x, one tries the places

$$h(x), \quad h(x) + c, \quad h(x) + 2c, \ldots$$

(Remember that '+' means addition with wraparound.)

As usual, m is the hash table size and c is any increment, so long as c is prime to m; otherwise the search will cycle through a fraction of the table.

We assume without loss of generality that $c = 1$.

We investigate the effect of inserting a random sequence

$$x_1, x_2, \ldots, x_n$$

of keys. With the key sequence left implicit, we define, for $0 \le j < m$,

$$B_j = \text{number of keys with hash-value } j$$
$$C_j = \text{number of keys whose search included } j \text{ and } j + 1$$

(C_j means 'the number of keys carried past j'.)

(13.2) Lemma *Bearing in mind that $j+1$ means $(j + 1) \mod m$,*

$$C_{j+1} = \begin{cases} 0 & \text{if } C_j + B_{j+1} \le 1 \\ C_j + B_{j+1} - 1 & \textit{otherwise.} \end{cases} \quad \blacksquare$$

Proof. We are considering the history of a series of insertions.

If $C_j = 0$ and $B_{j+1} = 0$, no key was carried past position j nor did any key hash to position $j + 1$, so there is no carry past $j + 1$.

If $C_j = 0$ and $B_{j+1} = 1$, no key was carried past position j and one key was stored at position $j + 1$, so there is no carry past $j + 1$.

If $C_j = 1$ and $B_j = 0$, there was one carried past j, reaching position $j + 1$, and nothing hashed to $j + 1$, so there is no carry past $j + 1$.

Let X be the set of keys carried past position j, and Y the set of keys hashed to position $j + 1$.

The only key from X which can end up at position $j + 1$ is the first. In this case all other keys from X, and all keys from Y, carry past $j + 1$.

If the first key to end up at position $j + 1$ was from Y, then every key in X, and every other key from Y, carries past $j + 1$.

Therefore $C_{j+1} = |X| + |Y| - 1$ where $|X| = C_j$ and $|Y| = B_{j+1}$. $\blacksquare$

(13.3) Let p_k be the terms in the Binomial$(n, 1/m)$ distribution, i.e., the probability that $B_j = k$, uniformly for any j. Let q_k the probability that $C_j = k$, uniformly for any j.

(13.4) Lemma

$$q_0 = p_0q_0 + p_0q_1 + p_1q_0$$
$$q_k = \sum_{r+s=k+1} p_rq_s \quad \textit{if } k \geq 1$$

Proof. $C_{j+1} = 0$ if and only if $C_j + B_{j+1} \leq 1$. Therefore

$$q_0 = p_0q_0 + p_0q_1 + p_1q_0$$

and $C_{j+1} = k > 0$ if and only if $C_j + B_{j+1} - 1 = k$, or for some r, s with $r + s = k + 1$,

$$C_j = r \quad \text{and} \quad B_{j+1} = s$$

whence

$$q_k = \sum_{r+s=k+1} p_rq_s. \quad \blacksquare$$

13.1 Generating functions

We assume a well-behaved hash function so that the probability that $h(x) = j$, $0 \leq j \leq m-1$, where x is a random key, is $1/m$.

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

Its generating function

$$B(z) = \sum_{k=0}^{n} p_kz^k =$$
$$\binom{n}{k}\left(\frac{z}{m}\right)^k(1 - \frac{1}{m})^{n-k} =$$
$$\left(1 + \frac{z-1}{m}\right)^n.$$

For purposes of calculation, we introduce another analytic function (polynomial) $D(z)$:

$$D(z) = \frac{B(z) - 1}{z - 1}$$
$$B(z) = 1 + (z-1)D(z)$$

Let $C(z)$ be the generating function for the q_j.
From the first line of Lemma 13.4

$$q_0z = p_0q_0z + (p_1q_0 + p_0q_1)z$$

and from the second line, if $k > 0$,

$$q_k z^{k+1} = \sum_{r+s=k+1} p_r q_s z^{k+1}$$

Expand $B(z)C(z)$

$$\begin{gathered}
B(z)C(z) = \sum_{k\geq 0} \sum_{r+z=k} p_r q_s z^k = \\
p_0 q_0 + \sum_{k\geq 1} \sum_{r+s=k} p_r q_s z^k = \\
p_0 q_0 + (p_0 q_1 + p_1 q_0) z + \sum_{k\geq 1} \sum_{r+s=k+1} p_r q_s z^{k+1} = \\
p_0 q_0 (1-z) + (p_0 q_0 + p_0 q_1 + p_1 q_0) z + \sum_{k\geq 1} q_k z^{k+1} = \\
p_0 q_0 (1-z) + q_0 z + \sum_{k\geq 1} q_k z^{k+1} = \\
p_0 q_0 (1-z) + zC(z) = B(z)C(z)
\end{gathered}$$

Now the ingenious substitution $B(z) = 1 + (z-1)D(z)$:

$$\begin{gathered}
p_0 q_0 (1-z) + zC(z) = C(z) + (z-1)D(z)C(z) \\
(z-1)C(z) - (z-1)C(z)D(z) = (z-1)p_0 q_0 \\
C(z)(1 - D(z)) = p_0 q_0 \\
C(z) = \frac{p_0 q_0}{1 - D(z)}
\end{gathered}$$

Since $C(1) = 1$ (true of all probability generating functions),

$$\frac{p_0 q_0}{1 - D(1)} = 1$$

so

$$\boxed{C(z) = \tfrac{1-D(1)}{1-D(z)}}$$

$$\begin{gathered}
C'(z) = \frac{(1 - D(1))(D'(z))}{(1 - D(z))^2} \\
C'(1) = \frac{D'(1)}{1 - D(1)}
\end{gathered}$$

$$\begin{gathered}
B(z) = 1 + (z-1)D(z) \\
B'(z) = D(z) + (z-1)D'(z) \\
B'(1) = D(1) \\
B''(z) = 2D'(z) + (z-1)D''(z) \\
B''(1) = 2D'(1)
\end{gathered}$$

$$\begin{aligned} B'(z) &= \frac{n}{m}\left(1+\frac{z-1}{m}\right)^{n-1} \\ B'(1) &= n/m \\ B''(z) &= \frac{n(n-1)}{m^2}\left(1+\frac{z-1}{m}\right)^{n-2} \\ B''(1) &= n(n-1)/m^2 \end{aligned}$$

Therefore

$$\begin{aligned} C'(1) = \frac{B''(1)}{2(1-B'(1))} &= \frac{n(n-1)/m^2}{2(1-n/m)} \\ &= \frac{n(n-1)}{2m(m-n)} \end{aligned}$$

13.2 Deriving S_n and U_n from $C'(1)$

Counting carries. Again we consider inserting keys $x_1, \ldots, x_n$ in that order. The key x_j will be inserted once it has been determined not to be already stored; the path followed in that initial unsuccessful search will be followed in any subsequent search for x_j.

Suppose that the path has length $r+1$, so r locations probed are already occupied. For each of these locations there is another carry-past, namely, x_j; so the search for x_j contributes to r of the carry-pasts.

The total number of probes is the total carry-past $+n$.

The *average* carry past $\sum_j jq_j$ at a fixed location is $C'(1)$. The total is $mC'(1)$, and therefore the average total number of probes is

$$mC'(1) + n = \frac{n(n-1)}{2(m-n)} + n. \tag{13.5}$$

(13.6) Definition *S_n and U_n are the average lengths of successful and unsuccesful searches with n items stored under linear rehashing.*

Divide Equation 13.5 by n to get S_n.

$$S_n = 1 + \frac{n-1}{2(m-n)}.$$

Splitting the constant,

$$S_n = \frac{1}{2} + \frac{n-1+m-n}{2(m-n)} = \frac{1}{2} + \frac{m-1}{2(m-n)}$$

Put $\alpha = n/m$, the hashing density.

(13.7) Lemma *The average length S_n of a successful search under linear rehashing is*

$$\frac{1}{2}+\frac{m-1}{2(m-n)}$$

Ignoring the -1 term, we get

$$S_n \approx \frac{1}{2}\left(1+\frac{1}{1-\alpha}\right). \quad ▮$$

To get the average length of an unsuccessful search, we observe that every key is added following an unsuccessful search in a table with fewer keys (clearly $U_0 = 1$), so

$$S_n = \frac{1}{n}\sum_{j=0}^{n-1} U_j.$$

so

$$\begin{aligned} U_n &= (n+1)S_{n+1} - nS_n \\ &= \frac{1}{2} + (n+1)\left(\frac{m-1}{2(m-n-1)}\right) - n\left(\frac{m-1}{2(m-n)}\right) \\ &= \frac{1}{2} + \left(\frac{m-1}{2}\right)\left(\frac{mn-n^2+m-n-mn+n^2+n}{(m-n-1)(m-n)}\right) \\ &= \frac{1}{2} + \frac{m(m-1)}{2(m-n-1)(m-n)}. \end{aligned}$$

By ignoring the -1 terms in the above expression, we get

(13.8) Lemma *The average length U_n of an unsuccessful search in a table with n items stored, linear rehashing, is approximately*

$$\frac{1}{2}+\frac{1}{2}\frac{1}{(1-\alpha)^2}. \quad ▮$$

13.3 Bibliographic notes

The analysis of linear rehashing is presented in 'Algorithms: their complexity and efficiency,' by Lydia Kronsjö.[1] Knuth (AOCP volume 3) gives a much more difficult analysis, and points out the slight inaccuracy in the other one (the 'carry past' probabilities are on an infinite probability space). That double hashing is close to uniform is apparently due to Guibas and Szemerédi, 1976.

[1] communicated by a student, Margaret Gallery