12 Open addressing: uniform and double hashing

Under open addressing, the keys are stored in the hash table. Convention: n is the number of keys stored and m is the size of the hash table. This time $n \leq m$ or else the table is full; as with chaining, the ratio n/m is significant.

(12.1) Definition $\alpha = n/m$.

The keys are stored in the table. It is necessary also that *empty* places in the table be recognised.

A problem, which does not arise when chaining is used, is that when searching for a key x with hash-value i, one may see table entries containing keys with *other* hash-values. In this case we have a so-called (secondary) *collision*.

12.1 Uniform hashing

(12.2) Definition *A hashing scheme has the* uniform hashing property *if the hash sequences are independent, so that search lengths follow a geometric distribution.*[1]

This means that in an unsuccessful search, the probability that an empty slot is found after r 'probes' is

$$\alpha^{r-1}(1-\alpha)$$

(12.3) Definition *U_n is the average cost of an unsuccessful search with n keys stored in the table. S_n is the average cost of a successful search.*

The estimate for U_n will use an infinite sum for simplicity. The average cost of an unsuccessful search is

$$\begin{aligned} U_n \quad &\approx \quad \sum_{r \geq 1} r\alpha^{r-1}(1-\alpha) = \\ &(1-\alpha)\frac{d}{d\alpha}\sum_{r\geq 1}\alpha^{r-1} = \\ &(1-\alpha)\frac{d}{d\alpha}\left(\frac{1}{1-\alpha}\right) = \quad \frac{1}{1-\alpha} \end{aligned}$$

To get S_n, a successful search with n keys stored matches an unsuccessful search with k keys stored, where $0 \leq k \leq n-1$. Write α_k for k/m and α for n/m.

$$\begin{aligned} S_n = \frac{1}{n}\sum_{k=0}^{n-1}\frac{1}{1-k/m} = \frac{m}{n}\sum_k \frac{1}{1-\alpha_k}\frac{1}{m} = \\ \frac{1}{\alpha}\sum_{k=0}^{n-1}\left(\frac{1}{1-\alpha_k}\right)\left(\frac{1}{m}\right) \end{aligned}$$

[1] It may be more correct to say hypergeometric. The difference is negligible.

or, if you prefer,

$$\frac{1}{\alpha}\sum_{k=0}^{n-1}\frac{1}{1-\alpha_k}\Delta\alpha_k$$

The sum can be approximated by the integral

$$\int_0^{\alpha}\frac{1}{1-x}dx = -\ln(1-\alpha)$$

Summarising

(12.4) Lemma *For uniform hashing,*

$$U_n \approx \frac{1}{1-\alpha}$$
$$S_n \approx \frac{1}{\alpha}\ln\frac{1}{1-\alpha} \quad \blacksquare$$

12.2 Double hashing

Uniform hashing is an ideal: double hashing approximates it. In double hashing, for every key x there are two hash functions, $h(x)$ and $k(x)$, and the search sequence is: a linear search beginning at $h(x)$ with steplength $k(x)$.

It is necessary that

$$\gcd(k(x), m) = 1$$

otherwise the search will cycle back before all locations have been probed.

(12.5) Proposition *Given independent random hash functions h and k, double hashing behaves like uniform hashing (bibliographic notes).* $\blacksquare$