Direct-Address Tables

Let $U = \{0, \ldots, m-1\}$ be the set of possible keys.
Use array $T[0 \ldots m-1]$ as a direct-address table.
There is a 1-1 correspondence between keys and slots.

DIRECT-ADDRESS-SEARCH(T, k)
  return $T[k]$

DIRECT-ADDRESS-INSERT(T, x)
  $T[key[x]] \leftarrow x$

DIRECT-ADDRESS-DELETE(T, k)
  $T[key[x]] \leftarrow$ NIL

Advantage: operations are $\Theta(1)$.
Disadvantage: $\Theta(|U|)$ space required.

Hash Tables

Let K be the set of keys to be stored.

Goal: use $\Theta(|K|)$ space and $\Theta(1)$ time/op.

Idea: Use array $T[0 \ldots m-1]$ as a hash table, and use a $\Theta(1)$ hash function h, where $h\colon U \rightarrow \{0, \ldots, m{-}1\}$ maps from keys to slots.

A *collision* is when two keys map to the same slot.

Good Hash Functions

Division method: $h(k) = k \bmod m$
m is prime, not close to any 2^i.

Division variation: $h(k) = (k \bmod M) \bmod m$
M is a big prime, not close to any 2^i.
m is any number much smaller than M.

Multiplication method:

$$h(k) = \lfloor m((kA) \bmod 1) \rfloor$$

m is a power of 2. $A = (\sqrt{5} - 1)/2$

Horner's Method for Division Hash Function

If $k = \langle k[1], \ldots, k[l] \rangle$, and if $0 \leq k[i] < r$, then compute hash function by:

$h \leftarrow k[1] \bmod m$
for $i \leftarrow 2$ **to** l
    do $h \leftarrow (rh + k[i]) \bmod m$

Universal Hashing

Let $\mathcal{H}$ be a set of hashing functions.
$\mathcal{H}$ is universal if $h(k) = h(k')$ with prob. $1/m$

m is a prime number.
$k = \langle k[1], \ldots, k[l] \rangle$, where $0 \leq k[i] < m$
Assign $a[i] \leftarrow \textsc{Random}(0, m-1)$

$$h(k) = \left(\sum_{i=1}^{l} a[i] * k[i] \right) \bmod m$$

The set of possible functions $h(k)$ is universal.
$h(k) = h(k')$ with prob. $1/m$. If $k[i] \neq k'[i]$,
$(a[i] * (k[i] - k'[i])) \bmod m$ has equally likely results.

Chaining

In chaining, slots are linked lists of the elements that hash to that slot, i.e., collisions.

Consider m slots, n elts., load factor $\alpha = n/m$.
Worst-case: $\Theta(n)$ if all elts. hash to same slot.
Best-case: $\Theta(1+\alpha)$, each slot has $\lfloor\alpha\rfloor$ or $\lceil\alpha\rceil$.

Average-case: Assume each slot is equally likely.

Unsuccessful search: $\Theta(1+\alpha)$
This is because average slot length $= \alpha$.

Successful search: $\Theta(1+\alpha)$
Before ith elt. inserted, avg. length $= (i-1)/m$.
Expected position of ith elt. $= 1 + (i-1)/m$.

Expected search length is the summation:

$\Sigma_{i=1}^{n}$	n elements to search for.
$1/n$	Prob. for ith element is $1/n$.
$1+(i-1)/m$	Expected position of ith elt.

$$\sum_{i=1}^{n} \left(\frac{1}{n}\right)\left(1+\frac{i-1}{m}\right) = 1 + \frac{\alpha}{2} - \frac{1}{2m}$$

Open-Address Hashing

In open addressing, when a collision occurs, probe for an empty slot and insert the new elt. there.

The hash function becomes:
$h : U \times \{0, \ldots, m-1\} \to \{0, \ldots, m-1\}$

The probe sequence $\langle h(k, 0), \ldots, h(k, m-1) \rangle$ should include all the slots.

```
HASH-INSERT(T, x)
    for i ← 0 to m − 1
        do j ← h(key[x], i)
            if T[j] = NIL
                then T[j] ← x
                    return j
    error "hash table overflow"
```

HASH-DELETE marks the slot as deleted.
HASH-SEARCH must continue past deleted slots.
HASH-INSERT can put new elts. in deleted slots.

Uniform Hashing Analysis

Uniform hashing assumes each probe sequence is equally likely.

Unsuccessful Search: $\Theta\left(\frac{1}{1-\alpha}\right)$
Let p_i = prob. exactly i probes find full slots.
Let q_i = prob. first i probes find full slots.
$p_i = q_i - q_{i+1}$
$q_1 = n/m = \alpha$ and $q_2 = \left(\frac{n}{m}\right)\left(\frac{n-1}{m-1}\right) < \alpha^2$

$$q_i = \prod_{k=0}^{i-1} \frac{n-k}{m-k} \le \left(\frac{n}{m}\right)^i = \alpha^i$$

Average number of probes is:

$$1 + \sum_{i=1}^{n} i\, p_i = 1 + \sum_{i=1}^{n} q_i \le \sum_{i=0}^{\infty} \alpha^i = \frac{1}{1-\alpha}$$

Successful Search: $\Theta\left(\frac{1}{\alpha} \ln \frac{1}{1-\alpha}\right)$
Inserting ith elt. = unsuccessful search $i-1$ elts.
Average number of probes is:

$$\sum_{i=1}^{n} \left(\frac{1}{n}\right)\left(\frac{1}{1-(i-1)/m}\right) \le \frac{1}{\alpha} \ln \frac{1}{1-\alpha}$$

Performance of Practical Methods

Linear Probing: $h(k,i) = (h'(k) + i) \bmod m$

Successful Search: $\Theta\left(\frac{1}{1-\alpha}\right)$

Unsuccessful Search: $\Theta\left(\frac{1}{(1-\alpha)^2}\right)$

Linear probing suffers from *primary clustering*, from long runs of occupied slots.

An empty slot preceded by i full slots gets filled next with probability $(i+1)/m$.

Quadratic Probing assumes m is a power of 2.

$$h(k,i) = (h'(k) + \frac{i}{2} + \frac{i^2}{2}) \bmod m$$

Successful Search: $\Theta\left(\frac{1}{\alpha} \ln \frac{1}{1-\alpha}\right)$

Unsuccessful Search: $\Theta\left(\frac{1}{1-\alpha}\right)$

Double Hashing, m is prime, $1 \le h_2(k) \le m-1$

$$h(k,i) = (h_1(k) + i\, h_2(k)) \bmod m$$

Successful Search: $\Theta\left(\frac{1}{\alpha} \ln \frac{1}{1-\alpha}\right)$

Unsuccessful Search: $\Theta\left(\frac{1}{1-\alpha}\right)$