Hash tables (III)

Textbook, Chapter 12, Sections 12.4 (end)

CSCE310: Data Structures and Algorithms

`www.cse.unl.edu/~choueiry/S01-310/`

Berthe Y. Choueiry (Shu-we-ri)

Ferguson Hall, Room 104

`choueiry@cse.unl.edu`, Tel: (402)472-5444

Open Addressing

Three techniques:

1. Linear probing
2. Quadratic probing
3. Double hashing

Main characteristics:

- All three guarantee that $\forall\, k$, $\langle h(k,1), h(k,2), \ldots, h(k,m) \rangle$ is a permutation on $\langle 0, 1, \ldots, m-1 \rangle$
- None fulfills the assumption of uniform hashing (since none is capable of generating more than m^2 probe sequences of the required $m!$)
- Double hashing has the greatest number of problem sequences and gives best results

Linear probing: Principle

Given a hash function: $h' \ : U \rightarrow \{0, 1, \ldots, m-1\}$
Linear probing uses: $h(k,i) = h(h'(k) + i) \bmod m$, for $i = 0, 1, \ldots, m-1$

Given k, probes: $T[h'(k)]$, $T[h'(k)+1]$, $T[h'(k)+2]$, …, $T[m-1]$, $T[0]$, $T[1]$, …, $T[h'(k)-1]$

- To insert an element: start with the hash value, proceed element by element until finding an empty slot
- To search for an element: start with the hash value, proceed element by element until finding the key sought

Example: *Courtesy of Dr.Cusack*
Let $h'(k) = k \bmod 13$, insert 18 41, 22, 59, 32, 31, 73 in h-table of size 13

Linear probing: Characteristics

- Uses only m distinct probe sequences
- Easy to implement
- Suffers from **primary clustering**: long runs of occupied slots build up, tend to get longer → Average search time increases
- Linear probing is not a good approximation to uniform hashing

Primary Clustering: Examples

- Table has $n = m/2$ keys stored.
  Every even-indexed slot is occupied, odd-indexed slot is empty
  $\Rightarrow$ Average unsuccessful search takes 1.5 probes
- If first $n = m/2$ first slots are the ones occupied, average number of probes becomes $n/4 = m/8$

Quadratic probing: Principle

Given a hash function: $h' : U \rightarrow \{0, 1, \ldots, m-1\}$
Quadratic probing uses: $h(k,i) = (h'(k) + c_1 i + c_2 i^2) \bmod m$
with c_1, c_2 auxiliary constants and $i = 0, 1, \ldots, m-1$

Given k, probes: $T[h'(k)]$, later positions are offset by amounts that depend in a quadratic manner on the probe number i

Quadratic probing: Characteristics

Quadratic probing uses: $h(k,i) = (h'(k) + c_1 i + c_2 i^2) \bmod m$

- Better performance than linear probing
- Initial probe determines entire sequence, only m distinct probe sequences are used
- To work well, values of c_1, c_2, m need to be selected carefully
- But, $h(k_1, 0) = h(k_2, 0) \Rightarrow h(k_1, i) = h(k_2, i)\ \forall\ i$:—(
- Yields a milder form of clustering: secondary clustering

Double hashing: one of the bests for open-addressing

Permutation produced have many of the characteristics of randomly chosen permutations

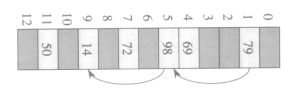
Double hashing uses: $h(k,i) = (h_1(k) + i h_2(k)) \bmod m$
where h_1, h_2 are auxiliary h-functions, and $i = 0, 1, \ldots, m-1$

Given k, probes: $T[h_1(k)]$, later positions are offset by amount of $h_2(k)$ modulo m
$\rightarrow$ probe sequence depends in two ways on the key, k
(1) initial probe varies with k
(2) offset value varies with k

Double hashing: Example I

Double hashing uses:
$h(k, i) = (h_1(k) + ih_2(k)) \bmod m$

using $h_1(k) = k \bmod 13$ and $h_2(k) = 1 + (k \bmod 11)$, insert $k = 14$



Double hashing: Characteristics

Double hashing uses: $h(k, i) = (h_1(k) + ih_2(k)) \bmod m$

- Ensure m and value of $h_2(k)$ are relatively prime (i.e., do not have a common divisor).
  Otherwise, for common divisor $d > 1$, search for a key will examine $(1/d)^{th}$ of h-table.
- One solution: $m = 2^p$ and h_2 always returns an odd number
- Another solution: m a prime number, and h_2 returns positive integer $< m$
  Example: m prime, $h_1(k) = k \bmod m$, and $h_2(k) = 1 + (k \bmod m')$, $m' = m - 1, m - 2$
- $\Theta(m^2)$ probe sequences are used
- Double hashing appears closer to the uniform hashing scheme

Double hashing: Example II *Courtesy of Dr.Cusack*

With $h_1(k) = k \bmod 13$, $h_2(k) = 1 + (k \bmod 8)$, and $m = 13$

We have: $h(k, i) = (h_1(k) + ih_2(k)) \bmod 13$

Insert the following keys: 18, 41, 22, 44, 59, 32, 31, 73

Double hashing: Performance

— Given $\alpha = n/m \leq 1$, average number of keys in h-table
— Assuming uniform hashing: the probe sequence $\langle h(k, 1), h(k, 2), \ldots, h(k, m)\rangle$ for each key k is equally likely to be any permutation on $\langle 0, 1, \ldots, m - 1\rangle$

Theorem 12.5: ... the expected numbers of probes is in an unsuccessful search is at most $1/(1 - \alpha)$

Corollary 12.6: Inserting an element requires at most $1/(1 - \alpha)$ probe on average

Theorem 12.7: ... the expected numbers of probes is in a successful search is $\frac{1}{\alpha} \ln \frac{1}{1-\alpha} + \frac{1}{\alpha}$.. and assuming every key in the table is equally likely to be searched for.