CSE 326
Lecture 20

Spring Quarter 1995

Closed Hashing (Open Addressing)

- General method of insertion: Let h: keys to {0,...,HSize - 1}

  1. Given a key x we first try to hash to h(x). This is the first "probe".

  2. If h(x) is occupied then we hash to (h(x) + f(1)) mod HSize. This is the second probe.

  3. If (h(x) + f(1)) mod HSize is occupied then we hash to (h(x) + f(2)) mod HSize. This is the third probe.

  Etc.

- The probe sequence is (h(x) + f(i)) mod HSize for i = 0,1,2,... assuming f(0)=0.

  If i reaches HSize -1 then the insertion fail, but it may happen for some f's that there is still space in the table.

  It is important to avoid long sequences of probes, so that load factors < 1/2 are commonly used in closed hashing.

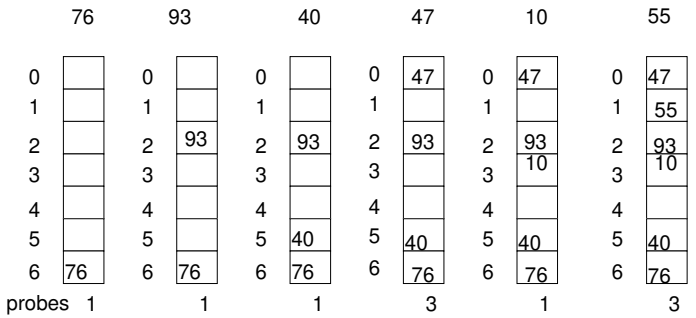
Linear Probing

- f(i) = i is linear probing
- Probe sequence is:
  h(x),
  (h(x) + 1) mod HSize,
  (h(x) + 2) mod HSize,
  (h(x) + 3) mod HSize, ...
- Linear probing insertion algorithm, assuming T is not full, i.e. $\lambda < 1$.

  compute h = h(x);
  while T(h) is not available do h := (h + 1) mod HSize;
  T(h) := x

Linear Probing Example

- h(x) = x mod 7



Performance of Linear Probing

- If there is an available slot then linear probing will find it.
- The expected number of probes on insertion is rougly

$$\frac{1}{2}\left(1 + \frac{1}{(1-\lambda)^2}\right)$$

Examples: $\lambda = .5$: expected number of probes is 2.5
$\lambda = .9$: expected number of probes is 50.5

- Linear probing suffers from primary clustering
- Do not use linear probing if load factor is > 1/2
- Linear probing or any other closed hashing method does not allow physical deletions. We must do lazy deletion.

Quadratic Probing

- $f(i) = i^2$ is quadratic probing
- Probe sequence is:
  h(x),
  (h(x) + 1) mod HSize,
  (h(x) + 4) mod HSize,
  (h(x) + 9) mod HSize, ...
- Note that the $h_{i+1} = h_i + 2i + 1$ where h_0 is the first probe, h_1 is the second probe and so on.
- Quadratic probing insertion is guaranteed to find an available slot provided the load factor is < 1/2 and HSize is prime.

Quadratic Probing Insertion Algorithm

- Assume assuming T has load factor < 1/2.

```
compute h = h(x);
if T(h) is not available then {
   i := 1;
   h := h + 1;
   while T(h) is not available do {
      h := (h + 2i + 1) mod HSize;
      i := i +1 }};
T(h) := x
```

- Example

h = h(50) = 50 mod 11 = 6
h = (h + 1) mod 11 = 7
h = (h + 3) mod 11 = 10
h = (h + 5) mod 11 = 4
h = (h + 7) mod 11 = 0

Quadratic Probing Finds a Slot if λ <1/2

- If HSize is prime then one of the slots $(h(x) + i^2)$ mod HSize is available for $0 \le i \le$ HSize/2.

  It is sufficient to argue that $(h(x) + i^2)$ mod HSize is different than $(h(x) + j^2)$ mod HSize for all i different from j with $0 \le i,j \le$ HSize/2.

  Suppose not. Then

  $h(x) + i^2 = h(x) + j^2$ (mod HSize)

  $i^2 - j^2 = 0$ (mod HSize)

  $(i - j)(i + j) = 0$ (mod HSize)

  But HSize does not divide either i - j or i + j.

Quadratic Probing may Fail if $\lambda > 1/2$

- 51 cannot be fit into this hash table with hash function h(x) = x mod 7.

0	
1	
2	16
3	45
4	59
5	
6	76

h(51) = 51 mod 7 = 2
(2 + 1) mod 7 = 3
(2 + 4) mod 7 = 6
(2 + 9) mod 7 = 4
(2 + 16) mod 7 = 4
(2 + 25) mod 7 = 6
(2 + 36) mod 7 = 3
(2 + 49) mod 7 = 2
.
.
.

Performance of Quadratic Probing

- Although quadratic probing can fail if $\lambda > 1/2$ it is not likely to do so. We can use load factors greater than 1/2, but, as usual, avoid load factors near 1.
- Quadratic probing does not suffer from primary clustering, but only has relatively minor secondary clustering.
- With load factors near 1/2 the expected number of probes per successful search is about 1.5.
- Again lazy deletion must be used.

Double Hashing

- f(i) = i $h_2(x)$ where h_2 is a second hash function.
- Probe sequence is:
  $h_1(x)$,
  $(h_1(x) + h_2(x))$ mod HSize,
  $(h_1(x) + 2\ h_2(x))$ mod HSize,
  $(h_1(x) + 3\ h_2(x))$ mod HSize,
- In choosing h_2 care must be taken that it never evaluates to 0, other wise there is no real probe sequence. One bad choice is $h_2(x) = x$ mod p.
- A good choice for h2 is to choose a prime R < HSize and let $h_2(x) = R - (x \bmod R)$.

Double Hashing Example

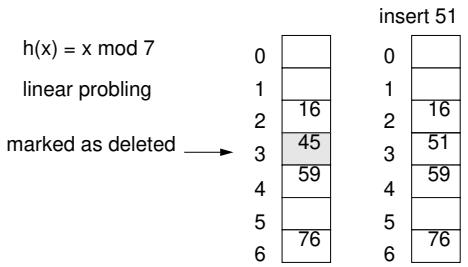
- $h_1(x) = x \bmod 7$ and $h_2(x) = 5 - (x \bmod 5)$

	76	93	40	47	10	55
0						
1				47	47	47
2		93	93	93	93	93
3					10	10
4						55
5			40	40	40	40
6	76	76	76	76	76	76
probes	1	1	1	2	1	2

Double Hashing is Safe for $\lambda < 1$

- Let $h_1(x) = x \bmod p$ and $h_2(x) = q - (x \bmod q)$ where $2 < q < p$ and q and p are primes. Then the probe sequence $(h_1(x) + i\, h_2(x)) \bmod p$ probes every entry of the hash table.
- Proof:

  Let $0 \leq k < p$. Let $h_1 = h_1(x)$ and $h_2 = h_2(x)$.
  We must show that $(h_1 + i\, h_2) \bmod p = k$ for some i.
  $0 < h_2 < p$ so that h_2 and p are relatively prime.
  By extended Euclid algorithm there exist s and t such that $s\, h_2 + t\, p = 1$.
  Choose $i = (k - h_1)\, s \bmod p$
  $(h_1 + i\, h_2) \bmod p =$
  $(h_1 + (k - h_1)\, s\, h_2) \bmod p =$
  $(h_1 + (k - h_1)\, s\, h_2 + (k - h_1)\, t\, p) \bmod p =$
  $(h_1 + (k - h_1)\, (s\, h_2 + t\, p)) \bmod p =$
  $(h_1 + k - h_1) \bmod p =$
  k

Deletion in Hash Tables

- Open Hashing (chaining) - Deletion is no problem
- Closed Hashing - Must do lazy deletion, that is, deleted items are marked as deleted.
  - Search is done normally in the various probing schemes.
  - Insertion - treat a deleted item as empty slot and fill it.