Hash Tables II
Chapter 5 in Weiss

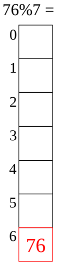
CSE 326
Data Structures
Ruth Anderson

Today's Outline

- **Announcements**
  - **Project 3 partner selection due Mon Feb 22 by 11pm, DO NOT WAIT UNTIL THEN TO START!**
  - **Written Homework #5 due Friday 2/19**
- **Today's Topics:**
  - **Hash Tables**
  - **Disjoint Sets**

Quadratic Probing Example

insert(76)	insert(40)	insert(48)	insert(5)	insert(55)
76%7 = 6	40%7 = 5	48%7 = 6	5%7 = 5	55%7 = 6



insert(47)
47%7 = 5

Student Activity

Quadratic Probing: Success guarantee for $\lambda < ½$

- If size is prime and $\lambda < ½$, then quadratic probing will find an empty slot in size/2 probes or fewer.
  - show for all `0 ≤ i,j ≤ size/2` and `i ≠ j`
    `(h(x) + i²) mod size ≠ (h(x) + j²) mod size`
  - by contradiction: suppose that for some i ≠ j:
    `(h(x) + i²) mod size = (h(x) + j²) mod size`
    `⇒ i² mod size = j² mod size`
    `⇒ (i² - j²) mod size = 0`
    `⇒ [(i + j)(i - j)] mod size = 0`

BUT size does not divide `(i-j)` or `(i+j)`

Quadratic Probing: Properties

- For *any* $\lambda < ½$, quadratic probing will find an empty slot; for bigger λ, quadratic probing *may* find a slot
- Quadratic probing does not suffer from *primary* clustering: keys hashing to the same *area* are not bad
- But what about keys that hash to the same *spot*?
  - ***Secondary Clustering!***

Double Hashing

f(i) = i * g(k)
where g is a second hash function

- Probe sequence:
  - 0th probe = h(k) mod TableSize
  - 1th probe = (h(k) + g(k)) mod TableSize
  - 2th probe = (h(k) + 2*g(k)) mod TableSize
  - 3th probe = (h(k) + 3*g(k)) mod TableSize
  - . . .
  - ith probe = (h(k) + i*g(k)) mod TableSize

Double Hashing Example

i^{th} probe = (h(k) + i*g(k)) mod TableSize

h(k) = k mod 7 and g(k) = 5 – (k mod 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

Resolving Collisions with Double Hashing

Hash Functions:

H(K) = K mod M

$H_2(K) = 1 + ((K/M) \bmod (M-1))$

M =

Insert these values into the hash table in this order. Resolve any collisions with double hashing:

13
28
33
147
43

Student Activity

Rehashing

Idea: When the table gets too full, create a bigger table (usually 2x as large) and hash all the items from the original table into the new table.

- When to rehash?
  - half full ($\lambda = 0.5$)
  - when an insertion fails
  - some other threshold
- Cost of rehashing?

Hashing Summary

- Hashing is one of the most important data structures.
- Hashing has many applications where operations are limited to find, insert, and delete.
- Dynamic hash tables have good amortized complexity.

Double Hashing Example

h(k) = k mod 7 and g(k) = 5 – (k mod 5)

0	
1	
2	
3	
4	
5	
6	