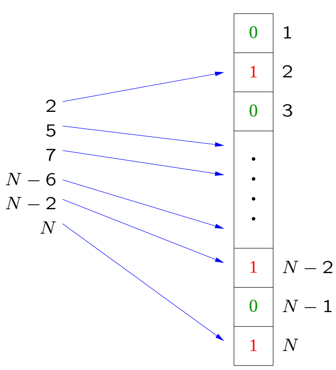
10—Hashing I

§8.3, 8.5

CSE326 Spring 2002

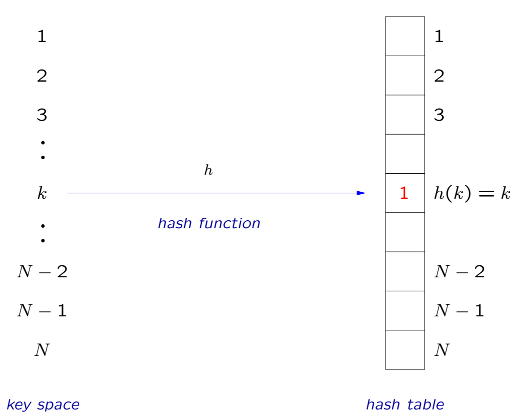
April 24, 2002

Easy Dictionary



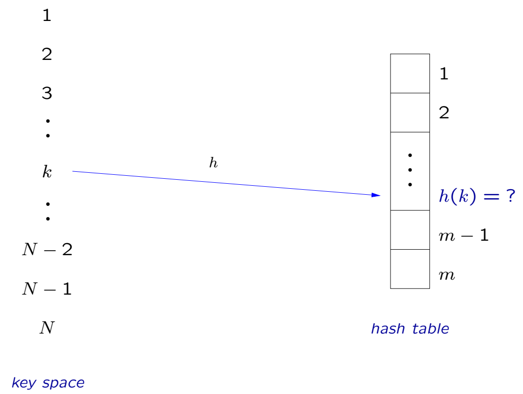
- n numbers in range $1 \ldots N$
- How long to *Find*?
- How long to *Insert*?
- How long to *Remove*?
- Why don't we use this?

Complicating the Issue



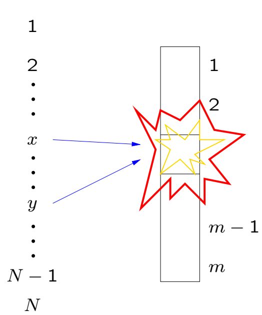
$$h : K \to T$$

Shrinking the Hash Table



What Do We Want From h?

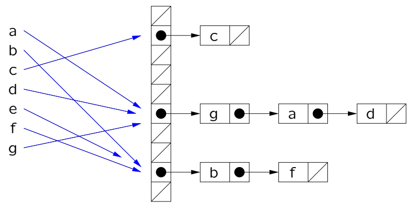
Comprehensive Collision Insurance



- $m < N$ so $h(x) = h(y)$ for some $x \neq y \in K$
- x and y *collide*
- How do we *resolve* the collision?
  - ⋆ *Chaining*
  - ⋆ *Open Addressing*

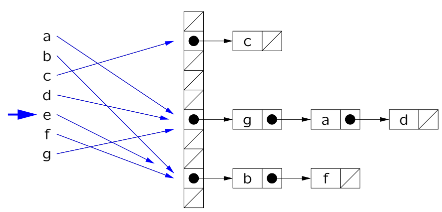
Chains of Love

Separate Chaining

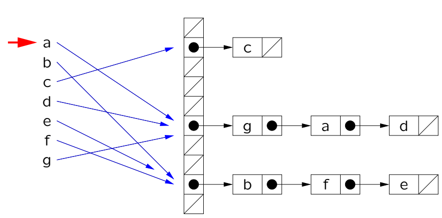


- T is a table of *buckets*
- How many *probes* to find c? d? e?

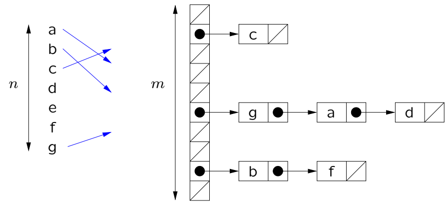
How to Insert?



How to Delete?

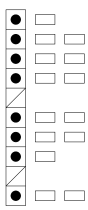


Important Quantities



- Load Factor: $\alpha = n/m$
- $S(\alpha)$: Expected # probes for *successful* search
- $U(\alpha)$: Expected # probes for *unsuccessful* search

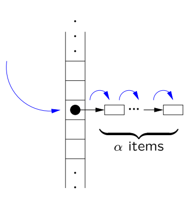
Motivating Assumption



$\alpha = 1.4$

- Keys *uniformly* distributed in T
- α accurately reflects lengths of chains
- S and U approximate *actual* performance
- What is worst-case?

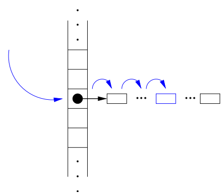
Calculating U



$$U(\alpha) = 1 + \alpha$$

$$U(\alpha) = E[1 + \text{chain search}] = 1 + E[\text{chain length}] = 1 + \alpha$$

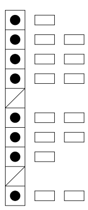
Calculating S



$S(\alpha) \approx 2 + \frac{\alpha}{2}$

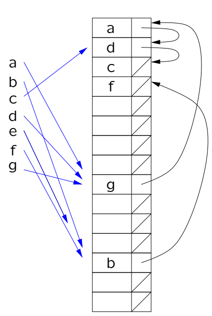
What should n and m be for constant-time operations?

Separate Chaining: Summary



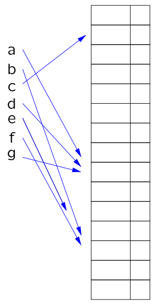
- Good performance if bucket lists aren't too long
  - ⋆ Both theoretically and practically
  - ⋆ Much depends on the hash function
- Same space overhead as binary tree
  - ⋆ key + pointer or two
  - ⋆ Still too much for many applications

Coalesced Chaining



Store buckets *internally* in the table

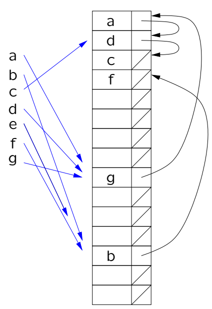
Coalesced Chaining



- Allocate new cells from top of table
- Insertion order:

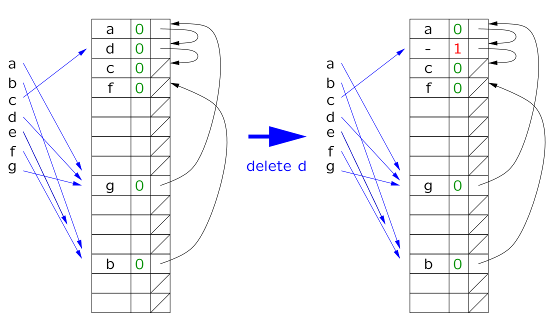
  g a d b c f

Coalesced Chaining Imperfect



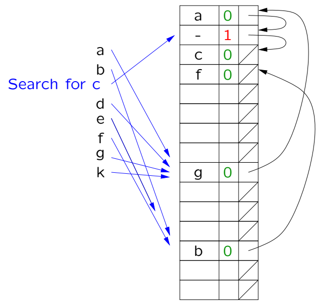
How to *Delete*?

Deleting



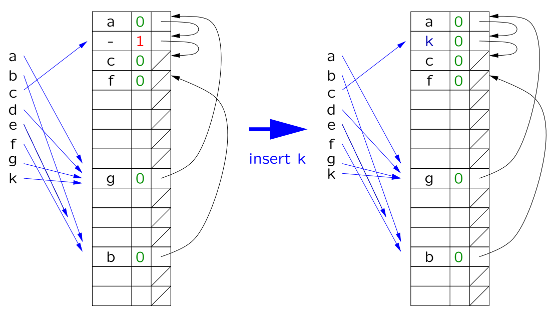
Add `deleted` field to entry

Searching After a Delete



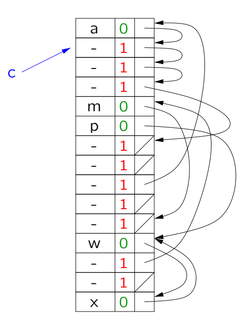
Skip over `deleted` = 1 entries

Inserting After a Delete



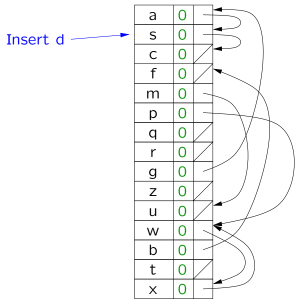
Watch `deleted` field when inserting

Delete Problems



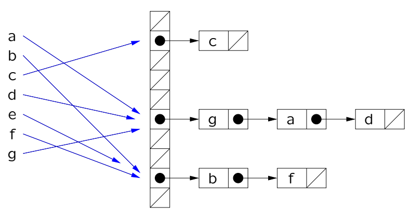
- What happens to searches with lots of deleted items?
- What happens to inserts?

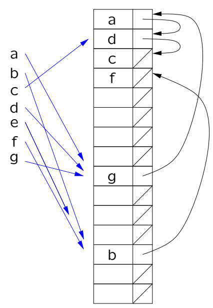
More Coalesced Problems



What happens when table gets full?

Open Addressing





How much space for Seperate Chained Table? Coalesced Chained?

- r-bit records (key + info), p-bit pointers

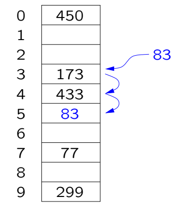
Open Addressing

0	450
1	
2	
3	173
4	433
5	
6	
7	77
8	
9	299

$h(k) = k$ mod 10

- Table is array of keys
- Need to set *probe sequence* to resolve collisions
  - ⋆ Linear Probing
  - ⋆ Quadratic Probing
  - ⋆ Double Hashing

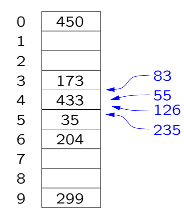
Linear Probing



$h(k) = k \bmod 10$

- First try $h(k)$...
- ...then $(h(k) + 1) \bmod m$
- ...then $(h(k) + 2) \bmod m$
- ...in general, $(h(k) + i) \bmod m$

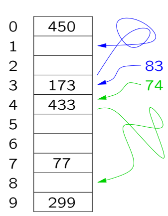
Problem



$h(k) = k \bmod 10$

Once a *primary cluster* gets started, it tends to grow and slow things down

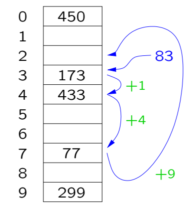
Problem Solution



$h(k) = k \bmod 10$

- This is a problem of the *probe sequence*
- We want:
  - ⋆ *Random* probe sequence (so no clusters form)
  - ⋆ *Deterministic* probe sequence (so we can find colliding keys)

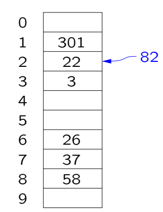
Quadratic Probing



$h(k) = k$ mod 10

- First try $h(k)$...
- ...then $(h(k) + 1^2)$ mod m
- ...then $(h(k) + 2^2)$ mod m
- ...in general,
  $(h(k) + i^2)$ mod m

Oops



$h(k) = k$ mod 10

Should be easy to insert 82, the table is only half full...

The Problem with Squaring

i	i mod 10
0	0
1	1
2	4
3	9
4	6
5	5
6	6
7	9
8	4
9	1

Maybe 10 was a bad choice for the table size?

Oops, We Did It Again

i	i mod 15
0	0
1	1
2	4
3	9
4	1
5	10
6	6
7	4
8	4
9	6
10	10
11	1
12	9
13	4
14	1

i	i mod 12
0	0
1	1
2	4
3	9
4	4
5	1
6	0
7	1
8	4
9	9
10	4
11	1

i	i mod 7
0	0
1	1
2	4
3	2
4	2
5	4
6	1

It's Not Just Us

- Just like regular math, $x^2 \equiv (-x)^2 \pmod{m}$
- $$-6\quad -5\quad \underbrace{-4\quad -3\quad -2\quad -1\quad \overbrace{0\quad 1\quad 2\quad 3\quad 4\quad 5}}_{\text{Another set of \#s modulo 10}}\overbrace{\quad 6\quad 7\quad 8\quad 9}^{\text{One set of \#s modulo 10}}\quad 10\quad 11$$
- $\cdots -8 \equiv 2 \equiv 12 \cdots \pmod{10}$

The World is Against Squares

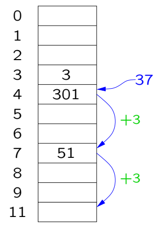
i	i mod 10
0	0
1	1
2	4
3	9
4	6
5	5
6	6
7	9
8	4
9	1

$\Rightarrow$

i	i mod 10
0	0
1	1
2	4
3	9
4	6
5	5
-4	6
-3	9
-2	4
-1	1

A quadratic probe sequence will *always* touch only *half* the table

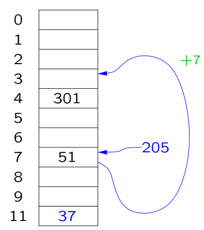
Double Hashing



$h(k) = k$ mod 11
$h_2(k) = \lfloor \frac{k}{11} \rfloor$ mod 11

- Probe sequence is $h(k) + i \cdot h_2(k)$
- 37 mod 11 = 4
- Probe increment $\lfloor \frac{37}{11} \rfloor$ mod 11 = 3

Double Hashing



$h(k) = k$ mod 11
$h_2(k) = \lfloor \frac{k}{11} \rfloor$ mod 11

- 205 mod 11 = 7
- Probe increment is $\lfloor \frac{205}{11} \rfloor$ mod 11 = 7