CHAPTER 8

Hashing

All the programs in this file are selected from

Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed
"Fundamentals of Data Structures in C /2nd Edition",
Silicon Press, 2008.

Symbol Table

- Definition

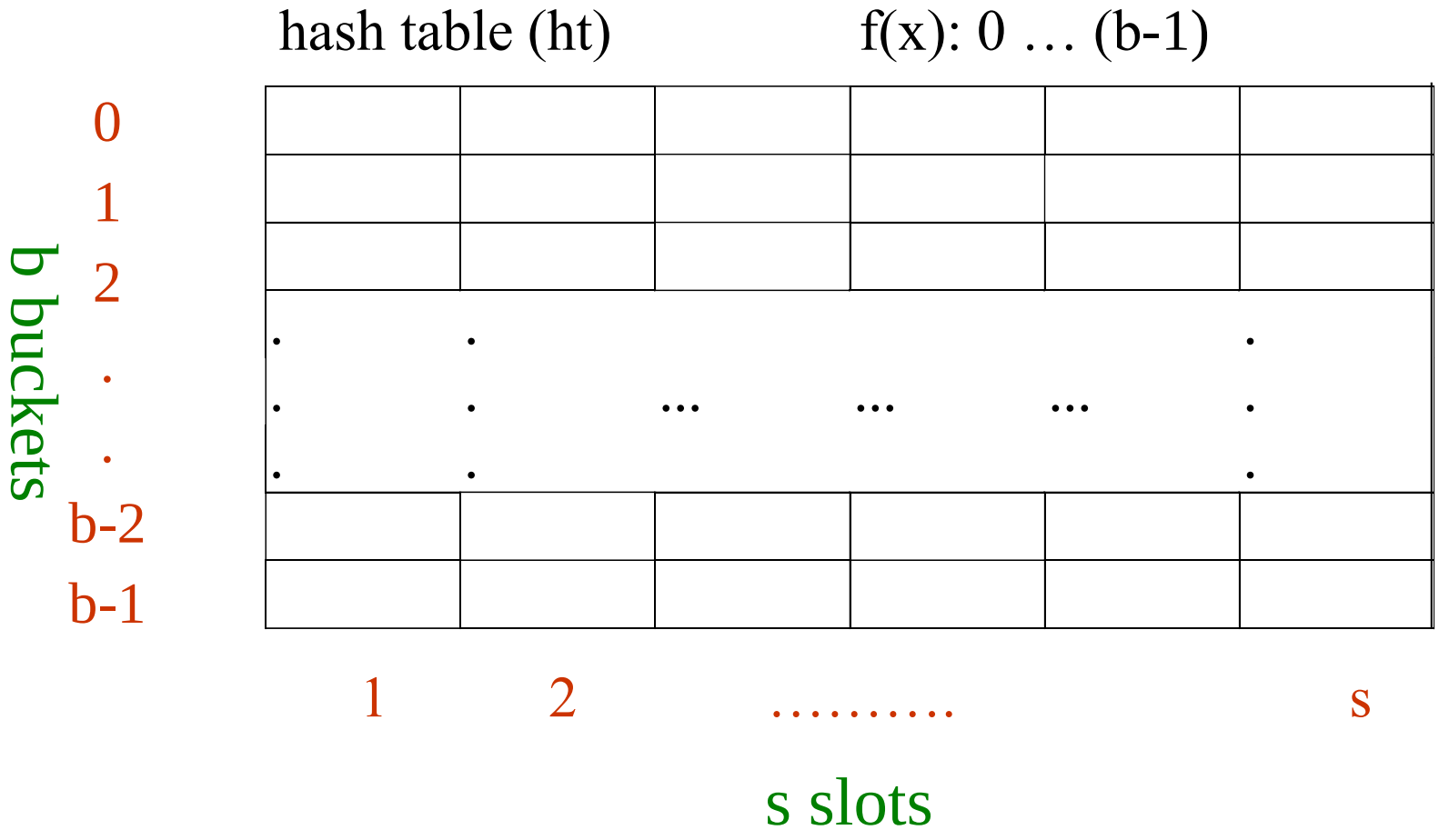
  A set of name-attribute pairs

- Operations
  - Determine if a particular name is stored in the table
  - Retrieve the attributes of the name
  - Modify the attributes of that name
  - Insert a new name and its attributes
  - Delete a name and its attributes

Search vs. Hashing

- Search tree methods: key comparisons
- hashing methods: hash functions
- types
  - static hashing
  - dynamic hashing

Static Hashing



Identifier Density and Loading Density

- The *identifier density* of a hash table is the ratio n/T
  - n is the number of identifiers in the table
  - T is possible identifiers
- The *loading density* or *loading factor* of a hash table is $\alpha = n/(sb)$
  - s is the number of slots
  - b is the number of buckets

Synonyms

- Two identifiers, *i* and *j* are synonyms with respect to *f* if *f*(*i*) = *f*(*j*)

Overflow and Collision

- An overflow occurs when we hash a new identifier into a full bucket
- A collision occurs when we hash two non-identical identifiers into the same bucket

Example

	Slot 0	Slot 1	
0	acos	atan	synonyms
1			
2	char	ceil	
3	define		
4	exp		
5	float	floor	synonyms
6			
…			
25			

synonyms:
char, ceil,
clock, ctime

overflow

b=26, s=2, n=10, α=10/52=0.19, f(x)=the first char of x

x: acos, define, float, exp, char, atan, ceil, floor, clock, ctime

f(x):0, 3, 5, 4, 2, 0, 2, 5, 2, 2

Hashing Functions

- Two requirements
  - easy computation
  - minimal number of collisions

1. Division

$$f_D(x) = x\%M \qquad (0 \sim (M-1))$$

2. mid-square (middle of square)

$$f_m(x) = middle(x^2)$$

Ex: 123^2=15126, 231^2=53361, 25^2=625

Avoid the choice of M that leads to many collisions

Example

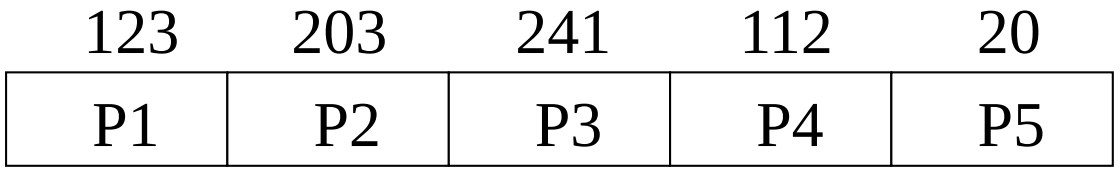
327%13=2
211%13=3

Identifier	Additive Transform	x	Hash
for	102+111+114	327	2
do	100+111	211	3
while	119+104+105+108+101	537	4
if	105+102	207	12
else	101+108+115+101	425	9
function	102+117+110+99+116+105+111+110	870	12

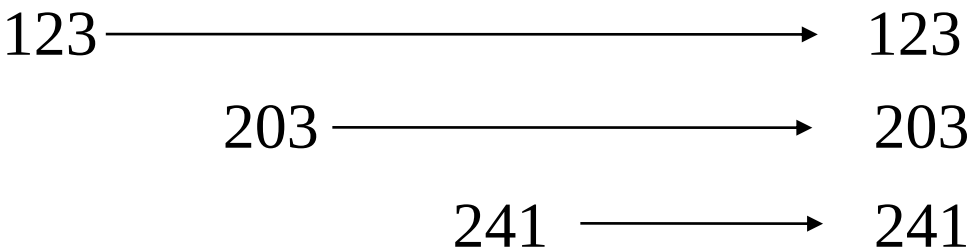
Hashing Functions

3. Folding
   - Partition the identifier x into several parts
   - All parts except for the last one have the same length
   - Add the parts together to obtain the hash address
   - Two possibilities
     - Shift folding
       - x1=123, x2=203, x3=241, x4=112, x5=20, address=699
     - Folding at the boundaries
       - x1=123, x2=203, x3=241, x4=112, x5=20, address=897

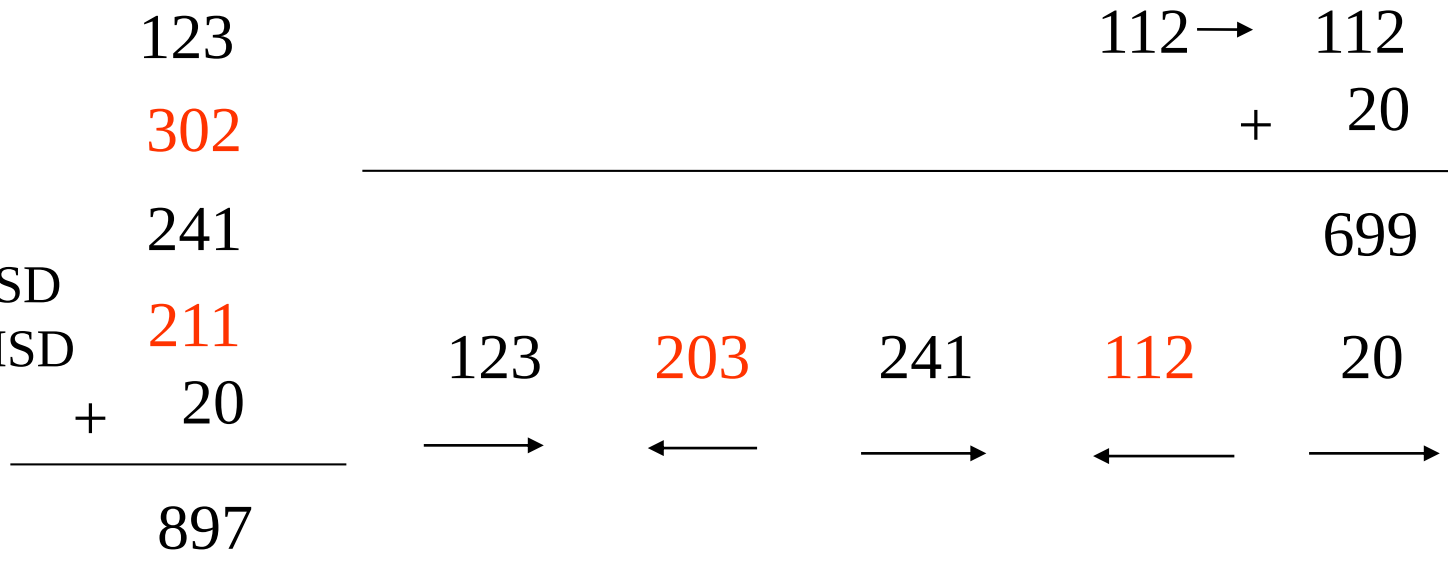
3-1 shift folding



3-2 folding at the boundaries



MSD ---> LSD
LSD <--- MSD



Digital Analysis

4. All the identifiers are known in advance

M=1~999

X_1 d_{11} d_{12} … d_{1n}

X_2 d_{21} d_{22} … d_{2n}

…

X_m d_{m1} d_{m2} … d_{mn}

1	2	3	4	5	6	7	8	9	679
5	2	1	4	3	6	1	7	5	615
5	2	1	2	6	7	2	6	4	724
5	2	1	2	6	8	3	6	3	833
5	2	1	4	3	9	4	8	8	948

Select 3 digits from n

Criterion:

Delete the digits having the most skewed distributions

Overflow Handling

1. Linear Open Addressing (linear probing)
2. Chaining

Linear Probing (linear open addressing)

- Compute f(x) for identifier x
- Examine the buckets

  ht[(f(x)+j)%TABLE_SIZE]

  $0 \le j \le$ TABLE_SIZE
  - The bucket contains x:Update
  - The bucket contains the empty string: Insert
  - The bucket contains a nonempty string other than x: Examine the next bucket
  - Return to ht[(f(x)+j)%TABLE_SIZE]: Error

acos, atoi, char, define, exp, ceil, cos, float, atol, floor, ctime
f(x)=first character of x

bucket	x	bucket searched	bucket	x	bucket searched
0 (a)	acos	1	1 (b)	atoi	2
2 (c)	char	1	3 (d)	define	1
4 (e)	exp	1	5 (f)	ceil	4
6 (g)	cos	5	7 (h)	float	3
8 (i)	atol	9	9 (j)	floor	5
10 (k)	ctime	9	…		
…			25 (z)		

Average number of buckets examined is 41/11=3.73

Problem of Linear Probing

- Identifiers tend to cluster together
- Adjacent cluster tend to coalesce
- Increase the search time

Improvement of Linear Probing

- Quadratic Probing
- Rehashing
- Random Probing

Hash Chaining

acos, atoi, char, define, exp, ceil, cos, float, atol, floor, ctime
f(x)=first character of x

[0] (a)	-> acos -> atoi -> atol
[1] (b)	-> NULL
[2] (c)	-> char -> ceil -> cos -> ctime
[3] (d)	-> define
[4] (e)	-> exp
[5] (f)	-> float -> floor
[6] (g)	-> NULL
…	
[25] (z)	-> NULL

# of key comparisons=21/11=1.91

α=n/b	.50	.75	.90	.95
hashing function	chain/open	chain/open	chain/open	chain/open
mid square	1.26/1.73	1.40/9.75	1.45/37.14	1.47/37.53
division	1.19/4.52	1.31/7.20	1.38/22.42	1.41/25.79
shift fold	1.33/21.75	1.48/65.10	1.40/77.01	1.51/118.57
Bound fold	1.39/22.97	1.57/48.70	1.55/69.63	1.51/97.56
digit analysis	1.35/4.55	1.49/30.62	1.52/89.20	1.52/125.59
theoretical	1.25/1.50	1.37/2.50	1.45/5.50	1.48