CS2x1:Data Structures and Algorithms

Koteswararao Kondepu

k.kondepu@iitdh.ac.in

Recap: Tree

(i) Tree;

(ii) Skew Tree

- Left Skew Tree
- Right Skew Tree

(iii) Binary Tree

- Full Binary Tree
- Perfect Binary Tree
- Complete Binary Tree
- Balanced Binary Tree

(iv) Binary Search Tree

- Create, Insert, Search, Delete
- Traversal (*Preorder, Inorder, Postorder*)
- Successor, Predecessor
- Minimum, Maximum

(v) AVL Tree

- Insert (*Left, Right, Left Right, Right Left Rotation)*
- Delete

(v) Priority Queue (heap)

- Insert
- Delete (*Minimum, Maximum*)
- Get (*Minimum, Maximum*)
- K^{th} - *smallest and largest*
- Sort and size

Tree time complexity:

→ O (n), where *n the number of nodes;*

– *worst case* → *skew tree*

→ *O (logn); balanced binary search tree*

→ *O (n); any tree travel orders*

List of Topics [C201]

- **Introduction:**
  - ***Data structures***
  - ***Abstract data types***
- **Creation and manipulation of linear data structures:**
  - ***Arrays; Stacks; Queues; Circular Queues; Singly Linked lists; Circular Singly Linked List; Doubly Linked List; Circular Doubly Linked List***
- **Introduction to Algorithms**
- Creation and manipulation of non-linear data structures:
  - ***Trees; Balanced trees [AVL]****;* ***Heaps;*** *Hash tables; Graphs.*
- Algorithms for sorting and searching, depth-first and breadth-first search, shortest paths and minimum spanning tree.

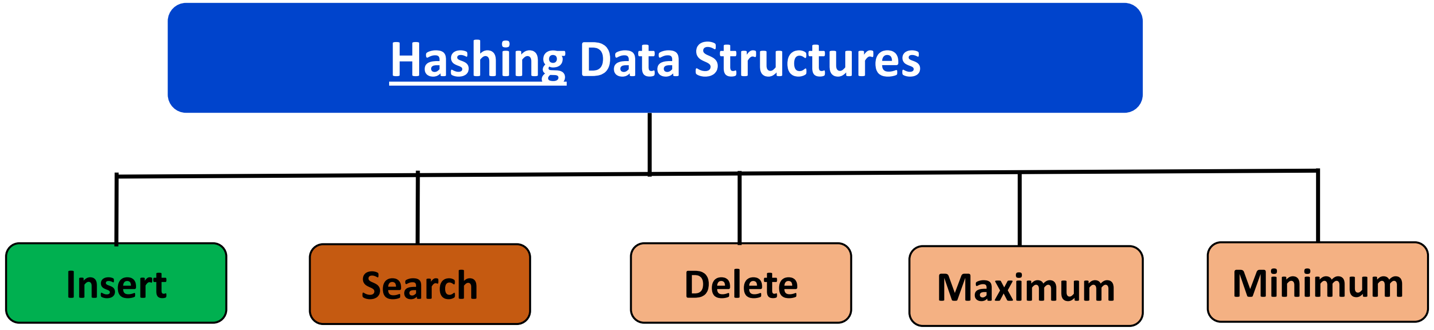
Motivation- Hashing

Tables

Let consider: Set of n distinct records with keys K_1, K_2, K_3, … K_n

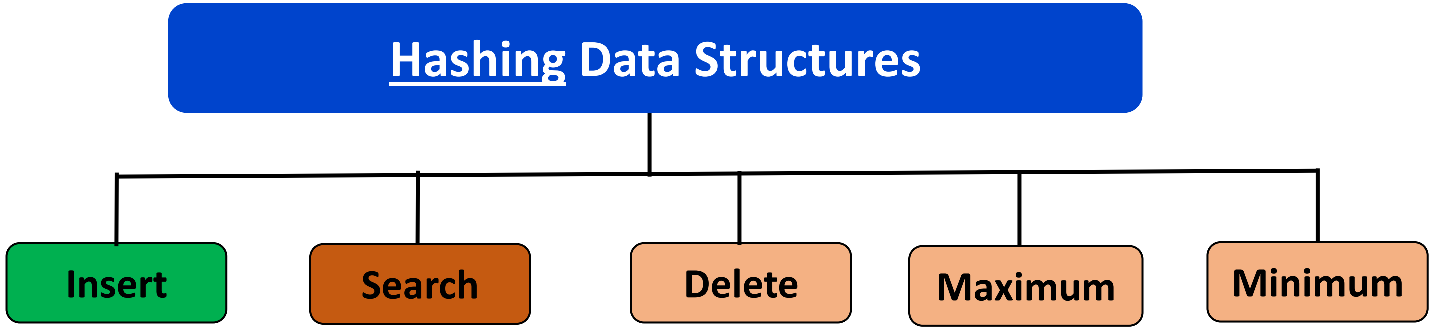
Find the record with the given Key values →

- *Sequential Search → Start from the first record and compare the key K with the keys of the record*
- *Searching time → the number of records in the file*



Hashing (2)

Tables



Elements of Hashing:

i. *Hash Table → contains the key values with pointers to the corresponding records*
ii. *Hash Function*
iii. *Collisions*
iv. *Collision Resolution Techniques*

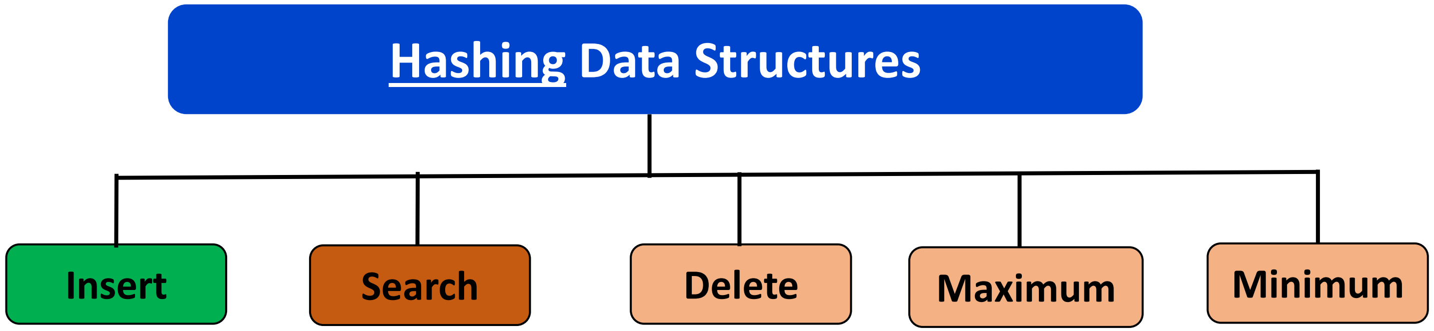
Hash Table

0	19
1	10
2	
3	49
4	59, 31, 77
5	
6	33
7	43
8	35, 62
9	

Hash Table → Collection of similar elements
Insert (S, x) → Modification of operation that arguments the Set S with the element x
Delete (S, x) → From the given pointer x to an element in the Set S, remove x from S
Search (S, k) → From the given a Set S and a Key k, return a pointer x to an element in S → key [x] == k or no if no element exists

Hashing (4)

Tables



Elements of Hashing:

- *i.* *Hash Table →contains the key values with pointers to the corresponding records*
- *ii.* *Hash Function*
- *iii.* *Collisions*
- *iv.* *Collision Resolution Techniques*

Hash tables are a very practical way to maintain a dictionary

Looking up an item in an array →O (1), once we know the index

Hash Function →a mathematical function which maps the key values to index values

Hash Table

0	19
1	10
2	
3	49
4	59, 31, 77
5	
6	33
7	43
8	35, 62
9	

Hash Function

Hash Function:

- *Add the two digits in the key*
- *After addition, take the digit at the unit place and ignore the digit at the tenth place, if any*

H: K → I

K	I
10	1
19	0
35	8
43	7
62	8
59	4
31	4
49	3
77	4
33	6

Hash Function

0	19
1	10
2	
3	49
4	59, 31, 77
5	
6	33
7	43
8	35, 62
9	

Hash Table

Collision

Collision: the set of keys mapped to the same index or more than one key value in one location in the hash table → ***collision***

|K| = |I| → the number of key values is the same as the size of the hash table

In general, |K| > |I|

0	19
1	10
2	
3	49
4	59, 31, 77
5	
6	33
7	43
8	35, 62
9	

Hash Table

H: K → I

K	I
10	1
19	0
35	8
43	7
62	8
59	4
31	4
49	3
77	4
33	6

Hash Function

Hash Function

A "good" hash function minimizes the probability of collisions

Hash Functions:

- *Division method*
- *Midsquare method*
- *Folding method*
- *Multiplication method*

Collision Resolution Technique:

- *Open addressing*
  - *Linear probing method*
  - *Quadratic probing method*
  - *Double hashing method*
- *Closed addressing*
  - *Chaining*

H: K → I

K	I
10	1
19	0
35	8
43	7
62	8
59	4
31	4
49	3
77	4
33	6

Hash Function

0	19
1	10
2	
3	49
4	59, 31, 77
5	
6	33
7	43
8	35, 62
9	

Hash Table

Hash Function: Division method

Most widely accepted hash function is the division method

H(k) = k mod m ; k → key, m → table size; if index start from 0
H(k) = k (mod m) + 1; if index start from 1

where k ∈ K

k % m = k mod m = Reminder by dividing k by m

K = 31; m = 13
H (31) = 31 mod 13 = 5

Recommendations: (i) select m → prime number or number without small divisors
(ii) equal to the size of the hash table

Exercise: Division method (1)

Consider a hash table H with 13 slots. The keys 10, 19, 35, 47, 63, 59, 31, 50, 77, and 33 are inserted into hash table using division method then what are the locations of keys 31, 47, and 50?

H(k) = k mod m ; k → key, m → table size; if index start from 0

K = 31; m = 13
H (31) = 31 mod 13 = 5

Select the following option for the locations of keys: 31, 47 and 50

A. 5, 7, 11
B. 5, 8, 11
C. 7, 5, 11
D. 8, 5, 11

Exercise: Division method (2)

Consider a hash table H with 13 slots. The keys 10, 19, 35, 47, 63, 59, 31, 50, 77, and 33 are inserted into hash table using division method then what are the locations of keys 31, 47, and 50, 11?

H(k) = k mod m ; k → key, m → table size; if index start from 0

K = 31; m = 13

H (31) = 31 mod 13 = 5

Select the following option for the locations of key

31, 47 and 50, 11:

A. 5, 7, 11, 11
B. 5, 8, 11, 11
C. 7, 5, 11, 11
D. 8, 5, 11, 11

H: K → I

K	I
10	10
19	6
35	9
47	8
63	1
59	4
31	5
50	11
76	
77	12
33	7

Hash Function

0	
1	63
2	
3	
4	59
5	31
6	19
7	33
8	47
9	35
10	10
11	50
12	77

Hash Table

Exercise: Division method (3)

Consider a hash table H with 10 slots. The keys 20, 30, 40, 50, 60, 70, and 80 are inserted into <u>hash table</u> using <u>division method!</u>

H(k) = k mod m ; k → key, m → table size; if index start from 0

K = 20; m = 10
H (20) = 20 mod 10 = 0

H: K → I

K	I
10	
20	
30	
40	
50	
60	
70	
80	

Hash Function

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Hash Table

Hash Function: Linear Probing

h'(k) = k mod m ; k → key, m → table size;
H(k, i) = (h' (k) + i) mod m; i → {0, 1, 2, ….m -1}

h'(k) → division or multiplication

if the index or location is filled in the hash table, the next linearly available index or location will be selected for inserting the keys in the hash table.

keys: 50, 700, 76, 85, 92, 73, and 101
m: 7

h'(50) = 50 mod 7 = 1
H(50, i=0) = (1+0) mod 7 = 1

h'(700) = 700 mod 7 = 0
H(700, i=0) = (0+0) mod 7 = 0

h'(76) = 76 mod 7 = 6
H(76, i=0) = (6+0) mod 7 = 6

h'(85) = 85 mod 7 = 1
H(85, i=0) = (1+0) mod 7 = 1 → ***Collision***
H(85, i=1) = (1+1) mod 7 = 2

h'(92) = 92 mod 7 = 1
H(92, i=0) = (1+0) mod 7 = 1 → ***Collision***
H(92, i=1) = (1+1) mod 7 = 2 → ***Collision***
H(92, i=2) = (1+2) mod 7 = 3

Hash Table

0	700
1	50
2	85
3	
4	
5	
6	76

Exercise: Linear Probing (1)

A hash table contains 10 buckets and uses *linear probing to resolve collisions*. The key values are integers and the *hash function h' (k) = k % 10*. If the values *43, 165, 62, 123, 142* are inserted in the table, in what location would the key value 142 be inserted?

$$H(k, i) = (h'(k) + i) \bmod m; \; i \rightarrow \{0, 1, 2, \dots m-1\}$$

$h'(43) = 43 \bmod 10 = 3$
$H(43, i=0) = (3+0) \bmod 10 = 3$

$h'(165) = 165 \bmod 10 = 5$
$H(165, i=0) = (5+0) \bmod 10 = 5$

$h'(62) = 62 \bmod 10 = 2$
$H(62, i=0) = (2+0) \bmod 10 = 2$

$h'(123) = 123 \bmod 10 = 3$
$H(123, i=0) = (3+0) \bmod 10 = 3$ → **Collision**
$H(123, i=1) = (3+1) \bmod 10 = 4$

Hash Table

0	
1	
2	62
3	43
4	123
5	165
6	
7	
8	
9	

A. 2
B. 3
C. 4
D. 6

Important:

$i \rightarrow \{0, 1, 2, \dots m-1\}$

Exercise: Linear Probing (1)

A hash table contains 10 buckets and uses *linear probing to resolve collisions*. The key values are integers and the *hash function h' (k) = k % 10*. If the values *43, 165, 62, 123, 142* are inserted in the table, in what location would the key value 142 be inserted?

Hash Table

0	
1	
2	62
3	43
4	123
5	165
6	
7	
8	
9	

$H(k, i) = (h'(k) + i) \bmod m; \; i \rightarrow \{0, 1, 2, \ldots m-1\}$

$h'(43) = 43 \bmod 10 = 3$
$H(43, i{=}0) = (3+0) \bmod 10 = 3$

$h'(165) = 165 \bmod 10 = 5$
$H(165, i{=}0) = (5+0) \bmod 10 = 5$

$h'(62) = 62 \bmod 10 = 2$
$H(62, i{=}0) = (2+0) \bmod 10 = 2$

$h'(123) = 123 \bmod 10 = 3$
$H(123, i{=}0) = (3+0) \bmod 10 = 3 \rightarrow$ **Collision**
$H(123, i{=}1) = (3+1) \bmod 10 = 4$

$h'(142) = 142 \bmod 10 = 2$
$H(142, i{=}0) = (2+0) \bmod 10 = 2 \rightarrow$ **Collision**
$H(142, i{=}1) = (2+1) \bmod 10 = 3 \rightarrow$ **Collision**
$H(142, i{=}2) = (2+2) \bmod 10 = 4 \rightarrow$ **Collision**
$H(142, i{=}3) = (2+3) \bmod 10 = 5 \rightarrow$ **Collision**
$H(142, i{=}4) = (2+4) \bmod 10 = 6$

A. 2
B. 3
C. 4
D. 6

Important:

$i \rightarrow \{0, 1, 2, \ldots m-1\}$

$i \rightarrow \{2, 4, 6, 8, 10, 12, 14, \ldots m-1\}$

Exercise: Division method (4)

A **<u>hash table of size 11</u>** using the hash function **<u>h(x) = x mod 11</u>** . The key values are given in the following order: 41, 27, 9, 21, 13, 22, 23, 26, 2, and 30.

*When we use <u>Linear Probing for collision resolution</u>, what is the index value of **key 30**, if the index value starts from 0 ?*

h'(k) = k mod m ; k → key, m → table size;

H(k, i) = (h' (k) + i) mod m; i → {0, 1, 2, ….m -1}

0	22
1	23
2	13
3	2
4	26
5	27
6	
7	
8	41
9	9
10	21

Hash Function: Quadratic Probing

h'(k) = k mod m ; k → key, m → table size;

H(k, i) = (h' (k) + i^2) mod m; i → {0, 1, 2, ….m -1}

h'(k) → division or multiplication

keys: 5, 56, 73, and 124
m: 17
i= starts from 0

h'(5) = 5 mod 17 = 5
*H(5, i=0) = (5+0*0) mod 17 = 5*

h'(56) = 56 mod 17 = 5
*H(56, i=0) = (5+0*0) mod 17 = 5 →* ***Collision***
*H(56, i=1) = (5+1*1) mod 17 = 6*

h'(73) = 73 mod 17 = 5
*H(73, i=0) = (5+0*0) mod 17 = 5 →* ***Collision***
*H(73, i=1) = (5+1*1) mod 17 = 6 →* ***Collision***
*H(73, i=2) = (5+2*2) mod 17 = 9*

h'(124) = 124 mod 17 = 5
*H(124, i=0) = (5+0*0) mod 17 = 5 →* ***Collision***
*H(124, i=1) = (5+1*1) mod 17 = 6 →* ***Collision***
*H(124, i=2) = (5+2*2) mod 17 = 9 →* ***Collision***
*H(124, i=3) = (5+3*3) mod 17 = 14*

Hash Table

0	
1	
2	
3	
4	
5	5
6	56
7	
8	
9	73
10	
11	
12	
13	
14	124
15	
16	

Exercise: Quadratic Probing (1)

Consider that the following Keys are inserted into a hash Table of size 10 using the <u>hash function h'(k) = k mod 10</u> *and Quadratic Probing is used for collision resolution.*

Keys: 9,19,29,39,49,59

What is the index into which 59 will be inserted?

$h'(k) = k \bmod m$; $k \rightarrow$ key, $m \rightarrow$ table size;
$H(k, i) = (h'(k) + i^2) \bmod m$; $i \rightarrow \{0, 1, 2, \ldots, m-1\}$

$h'(9) = 9 \bmod 10 = 9$
$H(9, i=0) = (9+0*0) \bmod 10 = 9$

$h'(19) = 19 \bmod 10 = 9$
$H(19, i=0) = (9+0*0) \bmod 10 = 9$ → **Collision**
$H(19, i=1) = (9+1*1) \bmod 10 = 0$

$h'(29) = 29 \bmod 10 = 9$
$H(29, i=0) = (9+0*0) \bmod 10 = 9$ → **Collision**
$H(29, i=1) = (9+1*1) \bmod 10 = 0$ → **Collision**
$H(29, i=2) = (9+2*2) \bmod 10 = 3$

$h'(39) = 39 \bmod 10 = 9$
$H(39, i=0) = (9+0*0) \bmod 10 = 9$ → **Collision**
$H(39, i=1) = (9+1*1) \bmod 10 = 0$ → **Collision**
$H(39, i=2) = (9+2*2) \bmod 10 = 3$ → **Collision**
$H(39, i=3) = (9+3*3) \bmod 10 = 8$

$h'(49) = 49 \bmod 10 = 9$
$H(49, i=0) = (9+0*0) \bmod 10 = 9$ → **Collision**
$H(49, i=1) = (9+1*1) \bmod 10 = 0$ → **Collision**
$H(49, i=2) = (9+2*2) \bmod 10 = 3$ → **Collision**
$H(49, i=3) = (9+3*3) \bmod 10 = 8$ → **Collision**
$H(49, i=4) = (9+4*4) \bmod 10 = 5$

Hash Table

0	19
1	
2	
3	29
4	
5	49
6	
7	
8	39
9	9

Exercise: Quadratic Probing (1)

Consider that the following Keys are inserted into a hash Table of size 10 using the hash function $h'(k) = k \bmod 10$ *and Quadratic Probing is used for collision resolution.*

Keys: 9,19,29,39,49,59

What is the index into which 59 will be inserted ?

$h'(k) = k \bmod m$; $k \rightarrow$ key, $m \rightarrow$ table size;

$H(k, i) = (h'(k) + i^2) \bmod m$; $i \rightarrow \{0, 1, 2, \ldots m-1\}$

$h'(9) = 9 \bmod 10 = 9$

$H(9, i=0) = (9+0*0) \bmod 10 = 9$

$h'(19) = 19 \bmod 10 = 9$

$H(19, i=0) = (9+0*0) \bmod 10 = 9$ → **Collision**

$H(19, i=1) = (9+1*1) \bmod 10 = 0$

$h'(29) = 29 \bmod 10 = 9$

$H(29, i=0) = (9+0*0) \bmod 10 = 9$ → **Collision**

$H(29, i=1) = (9+1*1) \bmod 10 = 0$ → **Collision**

$H(29, i=2) = (9+2*2) \bmod 10 = 3$

$h'(39) = 39 \bmod 10 = 9$

$H(39, i=0) = (9+0*0) \bmod 10 = 9$ → **Collision**

$H(39, i=1) = (9+1*1) \bmod 10 = 0$ → **Collision**

$H(39, i=2) = (9+2*2) \bmod 10 = 3$ → **Collision**

$H(39, i=3) = (9+3*3) \bmod 10 = 8$

$h'(49) = 49 \bmod 10 = 9$

$H(49, i=0) = (9+0*0) \bmod 10 = 9$ → **Collision**

$H(49, i=1) = (9+1*1) \bmod 10 = 0$ → **Collision**

$H(49, i=2) = (9+2*2) \bmod 10 = 3$ → **Collision**

$H(49, i=3) = (9+3*3) \bmod 10 = 8$ → **Collision**

$H(49, i=4) = (9+4*4) \bmod 10 = 5$

$h'(59) = 59 \bmod 10 = 9$

$H(59, i=0) = (9+0*0) \bmod 10 = 9$ → **Collision**

$H(59, i=1) = (9+1*1) \bmod 10 = 0$ → **Collision**

$H(59, i=2) = (9+2*2) \bmod 10 = 3$ → **Collision**

$H(59, i=3) = (9+3*3) \bmod 10 = 8$ → **Collision**

$H(59, i=4) = (9+4*4) \bmod 10 = 5$ → **Collision**

$\mathbf{H(59, i=5) = (9+5*5) \bmod 10 = 4}$

Hash Table

0	19
1	
2	
3	29
4	
5	49
6	
7	
8	39
9	9

Exercise: Division method (5)

A **hash table of size 11** using the hash function **h(x) = x mod 11** . The key values are given in the following order: 41, 27, 9, 21, 13, 22, 23, 26, 2, and 30.

When we use Quadratic Probing for collision resolution, what is the index value of key 30, if the index value starts from 0 ?

$h'(k) = k \bmod m$; $k \rightarrow$ key, $m \rightarrow$ table size;

$H(k, i) = (h'(k) + i^2) \bmod m$; $i \rightarrow \{0, 1, 2, \ldots m - 1\}$

0	22
1	23
2	13
3	2
4	26
5	27
6	
7	
8	41
9	9
10	21

Hash Function: Double hashing

$h_1(k) = k \bmod m$; $k \rightarrow key$, $m \rightarrow$ *table size;*
$h_2(k) = k \bmod m'$;

$h(k) = (h_1(k) + i * h2(k)) \bmod m; i \rightarrow \{0, 1, 2, \ldots m-1\}$

Insert the keys 79, 69, 98, 72, 14, 50 into the Hash Table of size 13. Resolve all the collisions using Double Hashing where the first and second hash functions are as specified above:

$$h(k) = (h_1(k) + i * h2(k)) \bmod m; i \rightarrow \{0, 1, 2, \ldots m-1\}$$

Hash Table

0	
1	79
2	
3	
4	69
5	
6	
7	98
8	
9	
10	
11	
12	

Hash Function: Midsquare

H(k) = x;
where x is obtained by selecting the appropriate number of bits or digits from the middle of the square of the key value k

Keys (k):	*1234*	*2345*	*3456*
k^2:	$1\underline{5}2\underline{2}7\underline{5}6$	$5\underline{4}9\underline{9}0\underline{2}5$	$11\underline{9}4\underline{3}9\underline{3}6$

Policy (selection criteria): select 3 digits at even positions from the right most digit in the square

H(k):	*525*	*492*	*933*

Keys (k):	*1234*	*2345*	*3456*
k^2:	$152\underline{2}756$	$549\underline{9}025$	$1194\underline{3}936$

Policy (selection criteria): select middle digit (r) bits or digits → range : 0 to $2^r - 1$

H(k):	*2*	*9*	*3*

thank you!

email:
k.kondepu@iitdh.ac.in