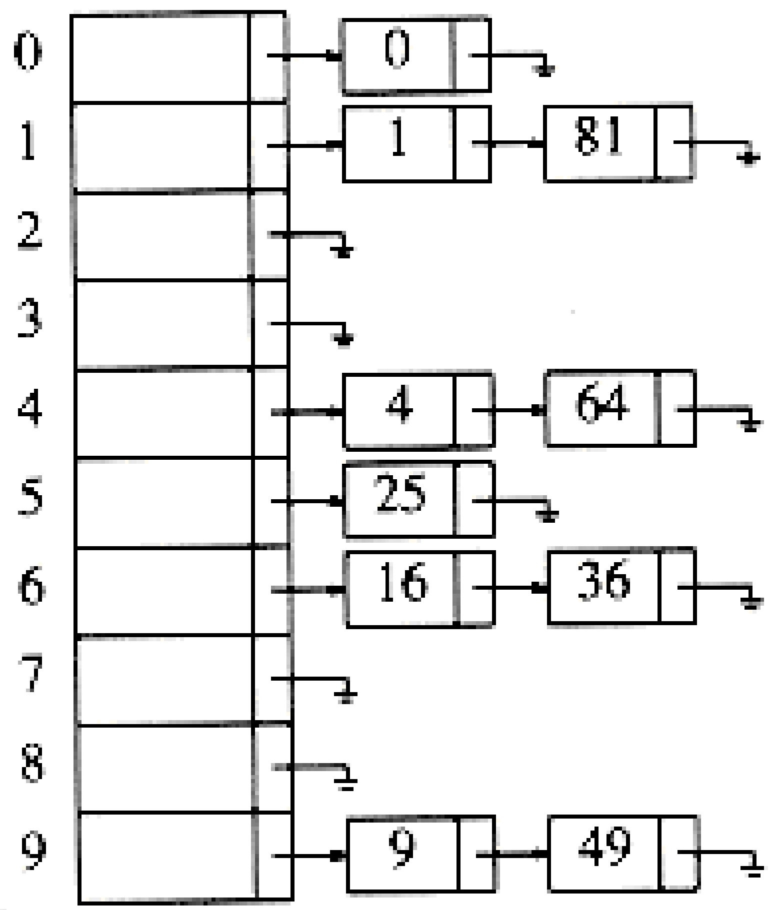
OPEN CLOSED HASHING

OPEN HASHING

Keep a list of all elements that hash to the same value.

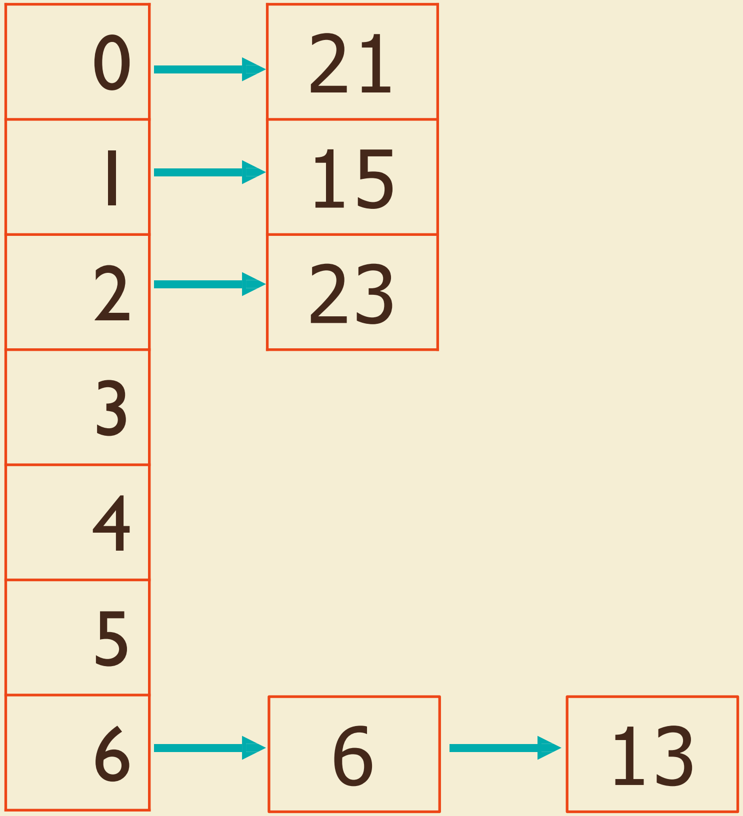
OPEN HASHING

Also known as
Separate Chaining


0
1
2
3
4
5
6
7
8
9
0
1
81
4
64
25
16
36
9
49

Insert 6, 15, 23, 21 and 13 to a hash table of size m = 7

key mod hSize



CLOSED HASHING

If a collision occurs, alternate cells are tried until an empty cell is found.

CLOSED HASHING

Also known as
Open Addressing

Closed Hashing

$h_0(x), h_1(x), \ldots$ are tried in succession where

$$h_i(x) = (hash(x) + f(i)) \bmod hSize$$

CLOSED HASHING

Linear Probing

Quadratic Probing

Double Hashing

Linear Probing

(hFunction(x) + f(i))%hSize

f is a linear function of i.

f(i) = i

i = number of collisions

Insert 89, 18, 49, 58 and 69 to a hash table of size $m = 10$

((key mod hSize) + i) mod hSize

i = number of collisions

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key mod hSize) + i) mod hSize

i = number of collisions

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key mod hSize) + i) mod hSize

i = number of collisions

0	
1	
2	
3	
4	
5	
6	
7	
8	18
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key mod hSize) + i) mod hSize

i = number of collisions

0	49
1	
2	
3	
4	
5	
6	
7	
8	18
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key mod hSize) + i) mod hSize

i = number of collisions

0	49
1	58
2	
3	
4	
5	
6	
7	
8	18
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key mod hSize) + i) mod hSize

i = number of collisions

0	49
1	58
2	69
3	
4	
5	
6	
7	
8	18
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key mod hSize) + i) mod hSize

i = number of collisions

0	49
1	58
2	69
3	
4	
5	
6	
7	
8	18
9	89

LINEAR PROBING

Primary Clustering

Any key that hashes into the cluster will require several attempts to resolve the collision, and then it will add to the cluster.

QUADRATIC PROBING

f is a quadratic function of i.

$$f(i) = c_1 * i + c_2 i^2$$

QUADRATIC PROBING

f is a quadratic function of i.

$f(i) = i^2$

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key mod hSize) + i^2) mod hSize

i = number of collisions

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key mod hSize) + i^2) mod hSize

i = number of collisions

0	
1	
2	
3	
4	
5	
6	
7	
8	18
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

$((\text{key mod hSize}) + i^2) \text{ mod hSize}$

i = number of collisions

0	49
1	
2	
3	
4	
5	
6	
7	
8	18
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size $m = 10$

((key mod hSize) + i^2) mod hSize

i = number of collisions

0	49
1	
2	58
3	
4	
5	
6	
7	
8	18
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key mod hSize) + i^2) mod hSize

i = number of collisions

0	49
1	
2	58
3	69
4	
5	
6	
7	
8	18
9	89

QUADRATIC PROBING

Secondary Clustering

Elements that hash to the same position will probe the same alternate cells.

DOUBLE HASHING

If a collision occurs, apply a second hash function to x.

DOUBLE HASHING

$f(i) = i * hash2(x)$

i = number of collisions

DOUBLE HASHING

$hash2(x) = R - (x\%R)$

R = prime smaller than hSize.

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key%hSize) +i*h2(key))%hSize

h2(key) = 7 - (key%7)

i = number of collisions

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key%hSize) +i*h2(key))%hSize

h2(key) = 7 - (key%7)

i = number of collisions

0	
1	
2	
3	
4	
5	
6	
7	
8	18
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key%hSize) +i*h2(key))%hSize

h2(key) = 7 - (key%7)

i = number of collisions

0	
1	
2	
3	
4	
5	
6	49
7	
8	18
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key%hSize) +i*h2(key))%hSize

h2(key) = 7 - (key%7)

i = number of collisions

0	
1	
2	
3	58
4	
5	
6	49
7	
8	18
9	89

Insert 89, 18, 49, 58 and 69 to a hash table of size m = 10

((key%hSize) +i*h2(key))%hSize

h2(key) = 7 - (key%7)

i = number of collisions

0	69
1	
2	
3	58
4	
5	
6	49
7	
8	18
9	89