HASHING

- Hashing or hash addressing is searching technique which is essentially independent of the number n.
- Assumation
  - (a) There is file F of n record with a set K of keys which uniquely determine records in F.
  - (b) F is maintained in memory by a table T of m memory locations & L is the set of memory addresses of the location in T.

Notations

F → File
n → records.
K → Keys. (it determine records in F)
T → table in memory in which F is maintained
m → memory location
L → set of memory addresses of location in T.

EXAMPLE

Suppose a company with 68 employee assign a 4-digit employee number to each employee which is used as the primary key in the company employee file.

F → company employe file
n → 68 (Records)
K → 4 digit employee number
L → Employee number as the address of record in memory.

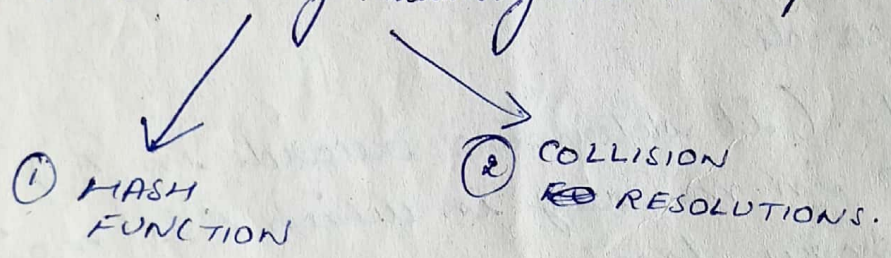
→ Modification of key (K) is done, so that great deal of space is not wasted. This modification takes the form of a function H from the set K of keys into the set L of memory addresses

$$H : K \to L$$

H → hash funct^n or hashing function

H (hash funct^n) doesnot yield distinct values i.e. it is possible that two different keys K_1 & K_2 will yield the same address & this situation is called collision

Division of Hashing in two parts



(1) HASH FUNCTION

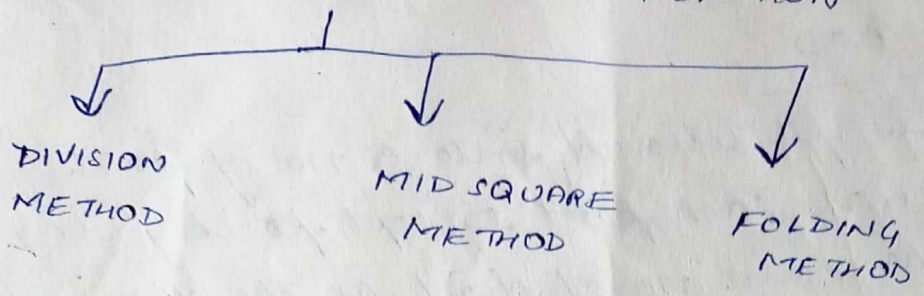
Selection of hash function as follows:

(a) H should be very easy & quick to compute

(b) Distribution of hash address should be uniform throughout L so that minimum collision.

→ General technique of hashing is to chop a key K into pieces & combine the pieces in some form to form hash address H(K).

TYPES OF HASH FUNCTION



(1) DIVISION METHOD / MODOLUS DIVISION METHOD

→ Choose a number 'm' larger than the no. n keys in K.
(m is generally prime number or a no. without small divisor)

FORMULA

(a) $H(K) = K \pmod m$ OR $K \% m$ [when range 0 to m-1 is required]

(b) $H(K) = K \pmod m + 1$ [when range 1 to m]

where
$K \pmod m$ → denotes the remainder when K is divided by m.

(2) MIDSQUARE METHOD

→ The key K is squared, then the hash function H is defined by

$$H(K) = l$$

→ where l is obtain by deleting digits from both ends of K^2.

Same position of K^2 must be deleted for all of the keys.

(3) FOLDING METHOD

→ The key K is partitioned into a no. of parts $K_1, \ldots K_r$ where each part except possibly the last, has the same no. of digit as the required address.

→ Then the parts are added together, ignoring the last ~~carry~~ carry i.e.

$$H(K) = K_1 + K_2 + \ldots + K_r$$

OR

Alternatively, one may reverse the second part before adding.

EXAMPLE :

A company with 68 employees is assigned a unique 4-digit employee number. Suppose L consist of 100 two-digit addresses: 00, 01, 02 --, 99. Apply hash function

3205, 7148, 2345

(a) DIVISION METHOD: (MODULOS – DIVISION METHOD)

$m = 97$ (∵ it is prime number close to 99)

$H(3205) = 3205 \% 97 = 4$ (3205/97 give reminder 4)

$H(7148) = 7148 \% 97 = 67$

$H(2345) = 2345 \% 97 = 17$

[In this case memory add° starting 00, so formula ① is used ② otherwise

5. MID-SQUARE METHOD

K =	3205	7148	2345
K^2	10 27 72 025	51 093 904	5 499 025
H(K):	72	93	90

C) FOLDING METHOD

Chopping the key K into two parts

H(3205) = 32+05 = 37

H(7148) = 71+48 = 19

H(2345) = 23+45 = 68

leading digit 1 in H(7148) is ignored

Alternatively, reverse the second part before adding, thus producing the foll. hash addresses

H(3205) = 32+50 = 82

H(7148) = 71+84 = 55

H(2345) = 23+54 = 77.

Example 2

→ consider a company with 68 employee. Each has been assigned a 4 digit employee number which is used as the primary key in the company's employee file. Suppose L (set of memory addresses of the locations) consist of 100 two digit addresses: 00, 01, 02, -- 99. Then apply the division method using a prime no. closest to 99, mid square method & folding method to find out the 2-digit hash add^{n} for each of the foll employ no. 9614, 5882, 1825

Division method

9614 → 9614 % 97 = 11

5882 → 5882 % 97 = 62

1825 → 1825 % 97 = 79

Mid square

K : 9614 5882 1825

K^2 : $(9614)^2$ $(5882)^2$ $(1825)^2$

92428996 34597924 3330625

28 97 06

Folding method

K: 9614 → 96+14 = 00

: 5882 → 58+82 = 30

1825 → 18+25 = 33

Reverse

96+41 = 37

58+28 = 76

18+52 = 60

1) DIRECT METHOD / LINEAR ADDR FUNCTION
2) MULTIPLICATION METHOD
3) HASHING
4) LENGTH DEPENDENT METHOD
5) PSEUDORANDOM METHOD

- DIVISION METHOD (1)
- MID SQUARE (2)
- FOLDING METHOD (3)
- SUBTRACTION METHOD (4)
- DIGIT EXTRACTION METHOD (5)
- ROTATION HASHING METHOD (6)

SUBTRACTION METHOD → IN SOME CASES KEYS ARE CONSECUTIVE BUT DO NOT START FROM 1. IN SUCH CASE WE SUBTRACT A NUMBER FROM KEY TO DETERMINE THE ADDR.

FOR. E.G (1) → IF A COMPANY HAS 200 EMPLOYEES, THEN THE EMPLOYEE NO. START FROM 2001, INSTEAD OF 1. IN THIS CASE SUBTRACTION HASHING USED, SUBTRACT 2000 FROM KEY TO DETERMINE ADDR

(2)

101	JANE
102	X
103	Y
104	Z
105	A
106	B

→ FIND ADDR OF 'Y'

→ SOLN → USING SUBTRACTION METHOD JUST SUBTRACT 100 FROM THE KEY 103. ∴ ADDR IS

(5)

→ Digit EXTRACTION METHOD — (a) SELECTED KEYS ARE EXTRACTED FROM THE KEY & MADE USE AS ITS ADDR USING A METHOD CALLED DIGIT EXTRACTION.

(b) IN THIS CASE, WE SELECT SPECIFIC DIGITS FROM KEY K & USE IT AS AN ADDR.

FOR E.G. (1) SUPPOSE WE WANT A HASH A 6 DIGIT EMPLOYEE NUMBER, 123457, TO A THREE DIGIT ADDR, WE COULD SELECT FIRST, THIRD, & FOURTH DIGITS FROM LEFT & USE THEM AS AN ADDR

SO ADDR WILL BE 124

QUES:

(2) SUPPOSE ROLL NO. OF A STUDENT WILLIAMS IS 160252. HASH THE NUMBER TO A THREE DIGIT ADDR USING DIGIT EXTRACTION METHOD.

ANS: SELECT FIRST, THIRD, & FOURTH DIGIT & USE IT AS THE ADDR

160252 → 102

∴ ADDR IS 102

(6) ROTATION HASHING METHOD →

→ THIS METHOD IS GENERALLY NOT USED BY ITSELF, BUT IS USED IN COMBINATION WITH OTHER HASHING METHODS. THIS METHOD IS ESPECIALLY USEFUL WHEN KEYS ARE ASSIGNED SERIALLY

→ E.G

ORIGINAL KEY		ROTATION		KEY AFTER ROTATION
500201		500201		150020
500202	→	500202	→	250020
500203		500203		350020
500204		500204		450020
500205		500205		550020

(7) LENGTH DEPENDENT METHOD

→ IN THIS METHOD LENGTH OF THE KEY IS USED ALONG WITH SOME PORTION OF THE KEY.

→ FOR E.G. KEY IS 12345 THE 123 IS TAKEN FROM KEY AND ADDED WITH LENGTH OF THE KEY i.e. 5

∴ ID IS 128

(8) PSEUDORANDOM METHOD

→ A random number is generated using the key. Using the modulo-division method a pointer is generated to the key. Formula to generate the pointer is

$P = (a * key + c) \% Tablesize$

where a → coefficient & c is constant.

(9) DIRECT METHOD OR LINEAR PROBING

→ The key is the address without any algorithmic manipulation.

→ Data structure must contain an element for every possible key.

E.G:

(10) Multiplication Method

→ Steps

1. Multiply the key by a constant A, $0 < A < 1$
2. Extract the fractional part of the product
3. Multiply this value by m.

$\therefore H(k) = FLOOR(m * (KA - FLOOR(KA))$

$A = .618033$ → GOOD CHOICE

E.G Ques: → 80 location

→ 2345, 5378, 7321

$H(2345) = FLOOR(79 * (2345 * .618033 - FLOOR(2345 * .618033)))$

$= FLOOR(79 * (1426.071885 - FLOOR(1426.071885)))$

$= FLOOR(79 * (1426.071885 - 1426))$

$= FLOOR(5.678915)$

$= 5$

$H(5378) = 61$ $H(7321) = 48$, $H(6911) = 17$

COLLISION

→ A collision is the event that occurs when hashing algorithm produce an address for an insertion key & that address is already occupied.

→ Add^{r} produce by hashing algorithm is known as home address

→ Prime area → memory that contains all of the home address.

→ When two keys collide at a home address, to resolve the collision by placing one of the key & its data in another location. Each calculation of an address & test for successive place is known as probe.

CLUSTERING

→ Some hashing algorithm tend to cause data to group within the list. This tendency of data to build up unevenly across a hashed list is known as clustering

→ TYPES OF CLUSTERING

PRIMARY CLUSTERING

→ It occurs when data become clustered around a home add^{r}

→ It is easy to identify

→ Random probing & quadratic probing

SECONDARY CLUSTERING

→ When data grouped along a collision path throughout a list

→ It is not easy to identify

→ It can be solved by second hashing function i.e. double hashing.

Collision Resolution Technique

→ To avoid collision.

(a) Spread out Records →

Finding a hashing algorithm that distributes the record fairly randomly among available addresses.

(b) Use Extra Memory →

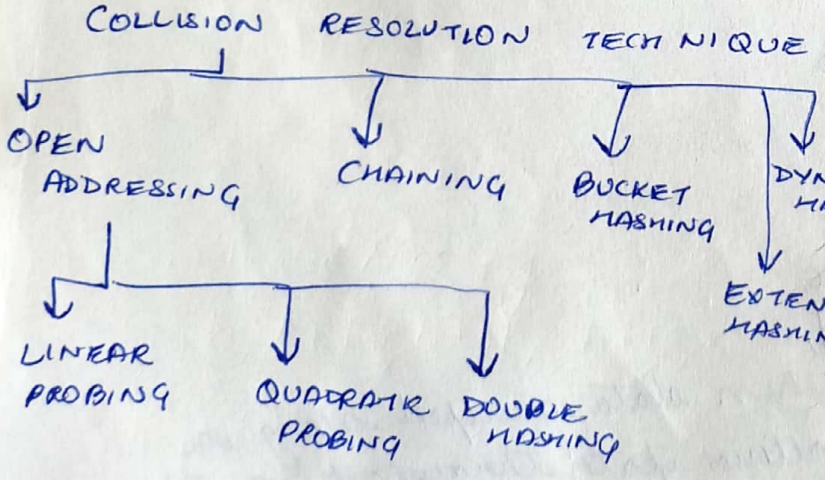
→ If there are more memory addresses to distribute the record
→ Then it is easier to find a hashing algorithm.

Adv: → Records are spread out evenly.

Disadv → waste spaces.

(c) Use Bucket →

→ Put more than one record at single add^{n}.



(a) Linear probing

→

Expected no. of probes using linear probing for insertion & unsuccessful search is

$$\frac{1}{2}\left(1+\frac{1}{(1-\lambda)^2}\right)$$

Successful search

$$\frac{1}{2}\left(1+\frac{1}{(1-\lambda)}\right)$$

where λ → load factor

Load factor of a hashed list is the no. of elements in the list divided by no. of physical elements allocated for the list expressed as percentage.

$\lambda = K/n * 100$

where n → table size
K → no. of element in table

(b) ~~Quadratic probe~~

Adv:

(a) Simple to implement
(b) Data tend to remains near the home address

Disadv.

(a) produce primary clustering.
(b) complicate search algo when data is deleted

Quadratic Probing

→ increment is collision probe no squared

→ $h+(1)^2, h+(2)^2, \ldots h+(i)^2$

Disadvantage

(a) Time required to square the probe

(b) It is not possible to generate new address for every element in the list.

Adv:

→ improvement over linear probing

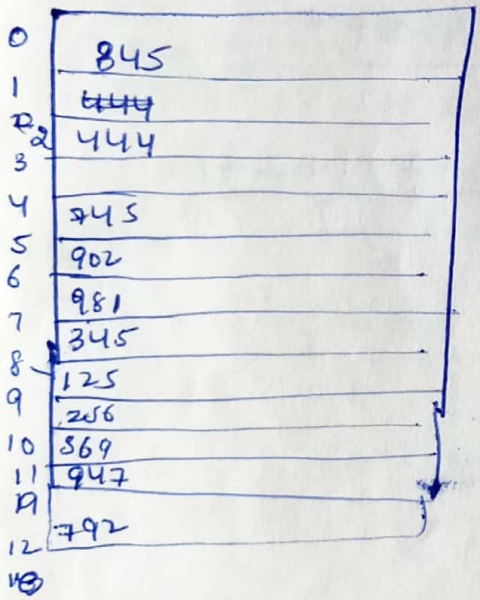
i.e. New addr is computed using another hash function instead of sequentially searching the hash table for an empty space.

Disadv:

→ Suffers from the displacement problem.

Double hashing

E.g



INSERT 652 into hash table

result in overflow at 2 we rehash

$h2(652) = 652 \bmod 11 = 3$ → NO. OF ELEMENTS

$rh(2, 652) = (2+3) \bmod 13 = 5$ already full

then rehash

$rh(5, 652) = (5+3) \bmod 13 = 8$

with space so store here.

Example

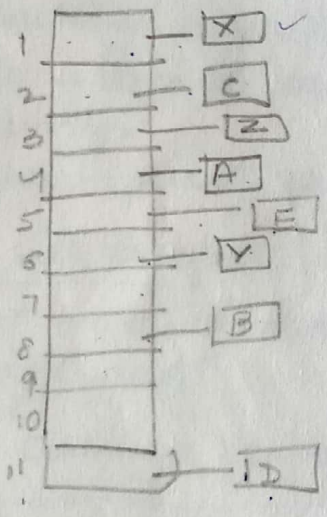
Ques: Suppose table T has 11 memory locations T[1], T[2], - - T[11] Suppose T consist of 8 records A, B, C, D, E, X, Y and Z with following hash addr.

RECORD: A, B, C, D, E, X, Y, Z
H(K): 4, 8, 2, 11, 4, 11, 5, 1

find S (Successful search probe) & U (Unsuccessful search probe) for

(a) Linear Probing (b) Quadratic Probing (c) Double hashing.

(a) LINEAR PROBING



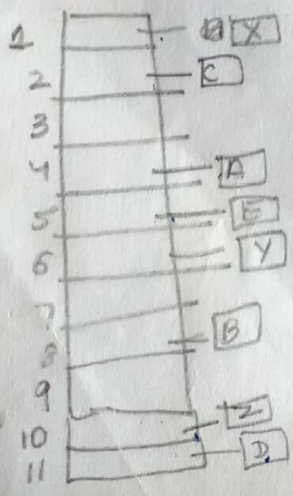
$S = \frac{1+1+1+1+2+2+2+3}{8} = \frac{13}{8} = 1.6$

$U = \frac{7+6+5+4+3+2+1+2+1+1+8}{11}$

$= \frac{40}{11} = 3.6$

(b) Quadratic Probing

$h, h+1, h+4, h+9, \ldots h+i^2$

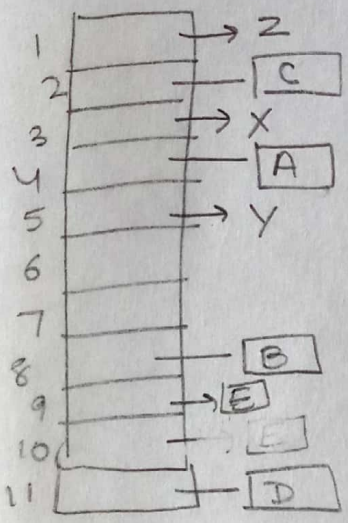


X →
E → h+1 = 5
X = h+1 = 1
Y = h,
Z = h, h+1, h+4, h+9
1, 1+1, 1+4, 1+9

$S = \frac{1+1+1+1+2+2+2+4}{8} = 14/8 = 1.75$

$U = (3+2+1+4+3+2+1+2+1+5+4)/11$

(c) Double hashing



INSERT E

→ $h' = 4 \bmod 8 = 51$

$h(5, 5) = (19) \bmod 11 = 9$

→ X →

$h' = 11 \bmod 8 = 3$

$h(3, 11) = 14 \bmod 11 = 3$

$$S = \frac{1+1+1+1+2+2+1+1}{8} = \frac{10}{8} = 1.25$$

$$U = \frac{6+5+4+3+2+1+1+3+2+1+7}{11} = \frac{35}{11} = 3.18$$