Hash Table

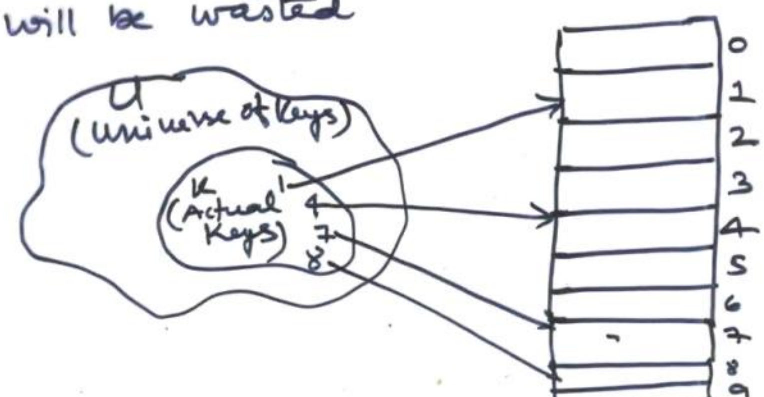
*→ A hash table in basic sense is a generalization of the simpler notation of an ordinary array.

*→ No. of employee = 68
Key = 4 digit employee No.

*→ can use employee No as the address of record in Memory.
No. of required location = 10000

*→ Actual Number of keys is very small as compared to the Total number of possible keys. Lots of space in memory will be wasted

*.



*→ Instead of using the key as array index the array index is computed from key itself.

H: U ⟶ K.
⟶ Hashing function.

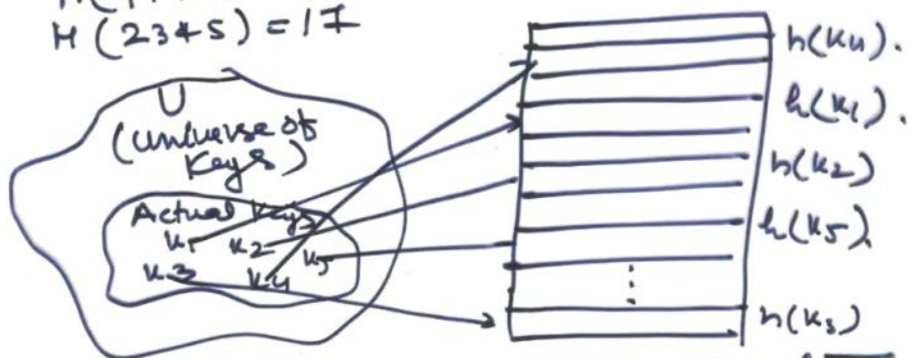
Example: No. of employee = 68
Key = 4 digit.
Array = 100.

$H(U) = U \bmod L$ ⟶ Assume it's 97

$H(3205) = 4$

$H(7148) = 67$

$H(2345) = 17$



*→ A hash table is a data structure in which the location of data item is determined directly as a function of data item itself

*→ direct address table.
Element with key K ⟶ slot k.

Hash table.
element with key K ⟶ slot h(k)

Hash function

→ It's simply a mathematical formulae that manipulate the key in some form to compute the index for this key in hash table

→ While designing a hash function.

*→ It should be possible to compute efficiently

*→ It should distribute the keys uniformly across the hash table. i.e It should keep the no. of collisions as minimum as possible.

Different Hash functions.

Division Method -:

$$h(k) = \underset{\text{Key}}{\underbrace{k}} \bmod \underset{\text{No. of slots in Hash table.}}{\underbrace{m}}$$ { if zero indexing is allowed

$$h(k) = k \bmod m + 1$$ { otherwise }

Example: $m = 9$.
$h(k) = k \bmod m$ will map the key 132 to slot 7.

$$h(132) = 132 \bmod 9$$
$$= 7$$

Note: In using the division method, certain values of m should be avoided. It has been proved in literature that good values ~~for m~~ are prime values not too close to exact power of 2.

— {Multiplicative Method}

$h(k) = \lfloor m\,(kA \bmod 1) \rfloor$

→ k is multiplied by a constant A in the range $0 < A < 1$ and extract fractional part of kA

→ choice of A generally used by users = 0.618

(Proof by Knuth
$A \simeq (\sqrt{5}-1)/2 = 0.618$)

Example: Consider a hash table with 10000 slots, i.e. $m = 10000$. Then the hash function $h(k) = \lfloor m\,(kA \bmod 1) \rfloor$ will map 123456 to 41

- { Mid Square Method }

Step 1 : square the key value k is taken

step 2 : hash value is obtained by deleting digits from end of the squared value.

Ex : consider a hash table with 100 slots. i.e m=100 and key value. K = 3205, 7148, 2345

k:	3205	7148	2345
k^2 =	10272025	51093904	5499025
h(k) =	72	93	90

- { folding method }

step 1 : key value k is divided into number of parts $k_1, k_2, \ldots k_3 \ldots k_r$. where each part has the same. Number of digits except the last part

step 2 : These parts are added together and the hash value is obtained by ignoring the last carry.

Ex : consider a hash table with 100 slots. i.e m = 100 and following key values

k =	9235	714	71458
parts =	92, 35	71, 4	71, 45, 8.
sum =	127	75	114
h(k) =	27	75	14

5

→ {Resolving collision}

following schemes are used.

→ separate chaining

→ open addressing.

– {Collision Resolution by separate chaining}

→ All the Elements whose keys hash to the same hash table slot are put in a one linked list

→ slot i in the hash table contains a pointer to the head of the linked list of all the elements that hash to value i, if there are no such element that hash to value i, the slot i contains NULL value.

Ex: insert the elements 5, 28, 19, 15, 20, 33, 12, 17, 10 into a chained hash table, Assume hash table has 9 slots and hash function be $h(k) = k \bmod 9$.

Collision Resolution by open Addressing.

→ If the slot to which key is hashed is free then the element is stored in that slot

→ If it's filled, the next free slots are identified systematically i.e forward direction.

→ Process of examining the slots in the hash table is called probing.

- → linear Probing
- → quadratic probing
- → double Hashing.

– {Linear Probing} :-

$$h(k, i) = [h'(k) + i] \bmod m.$$

i → Probe No.

for $i = 0, 1, 2, \ldots m-1$.

Example: Consider inserting the keys 76, 26, 37, 59, 21, 65, 88 into a hash table of size $m = 11$.

T =

0	1	2	3	4	5	6	7	8	9	10
21	65	88		26	37	59				76

Quadratic probing.

$h(k,i) = [h'(k) + c_1 i + c_2 i^2] \bmod m.$

where $h'(k) = k \bmod m$

Example: consider inserting the keys 76, 26, 37, 59, 21, 65, 88 into a hash table of size $m = 11$ using quadratic probing with $c_1 = 1$ and $c_2 = 3$.

consider $h'(k) = k \bmod m$

Double Hashing:

$h(k,i) = [h_1(k) + i\, h_2(k)] \bmod m.$

$h_1(k) = k \bmod m$

$h_2(k) = k \bmod m'.$

m' is chosen to be slightly less than m (say $m-1$ or $m-2$)

Ex: consider inserting the keys 76, 26, 37, 59, 21, 65, 88 into a hash table of size $m = 11$ using double hashing, consider

$h_1(k) = k \bmod 11$

$h_2(k) = k \bmod 9.$