**HASHING**

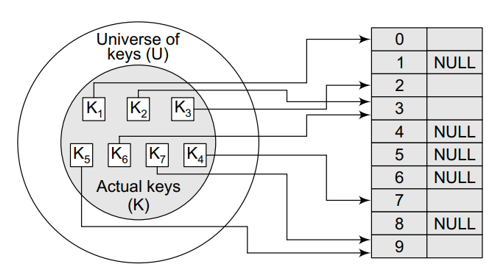
**Introduction**

* Hash table = Hash map
* Has **unique** key.
* Pairs are **randomly** distributed.

**Usecase**

* Database
* Caching mechanism
* Compilers
* Sets/ dictionaries

**Index-Key Relationship**

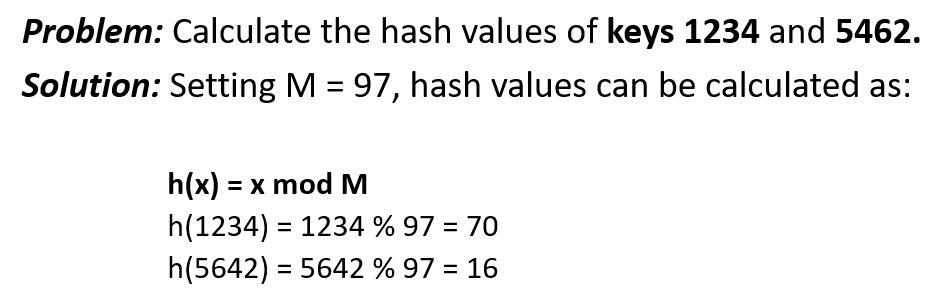


**Hash Function**

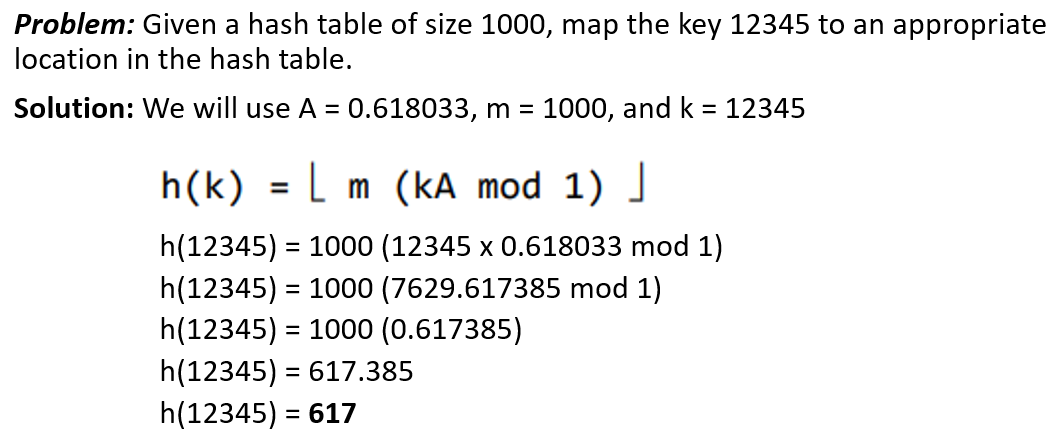
* Are mathematical formulae used in hashing.
* **Collision:** Hash keys storing identical values.
* Good hash table = Less collision
* Less collision is achieved by **uniform** pair distribution.
* **Hash value:** Unique numeric value identifying data.

**Hash Function Methods**

* Division method **[h(x) = x % M]**:-



* **Multiplication method:-**



(Value of A is suggested by Knuth)

**Other Hashing Techniques**

* MurmurHash (3 variants)
* CityHash
* Secure Hash Algorithm – 1
* Secure Hash Algorithm 256
* Cyclic Redundancy Check 32
* Universal Hashing
* Fowler Noll Vo

**Properties of Good Hash Function**

* Low cost (fast algorithm)
* Determinism (can determine based on conditions)
* Uniformity (low collision)
* Bucketed (storing same keys in same bucket)

**Terminologies**

* **Probing:** Techniques used in avoiding collisions by finding alternative slots.
* **Hashing:** Encryption of values into certain unique string or number.

**Collision Handling Techniques**

* **Chaining:** Each bucket contains a linked-list containing pairs with same key value.
* **Open addressing:** Algorithm searches for available slot via **probing**.
* **Robin hood hashing:** Balances bucket lengths moving **heavy to light** ones.
* **Cuckoo Hashing:** Stores keys in one of the many possible locations.

**Load Factor**

* Shows how much a hash table is filled.
* Ratio of: **Total key-value pairs: Total buckets**
* High load = Collisions + Bad performance
* Low load = Memory wastage
* Hash-map sizes are reshaped when it reaches threshold.

**Resizing Hash Tables**

* Hash-maps are resized as said before.
* Rearranges hashes into new buckets.
* Costly but effective.

**Nature of Hash-Maps**

* Non-linear as linear functions have clear order/ sequence.

**Static & Dynamic Hashing**

* **Static hashing properties:-**
  + Fixed size throughout life
  + No auto-resizing
  + Predictable
* **Dynamic hashing properties:-**
  + Variable size
  + Balanced load
  + More efficient