SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA
THANJAVUR | KUMBAKONAM | CHENNAI

I B.Tech [CSBS]
CSE209 - Data Structures & Algorithms

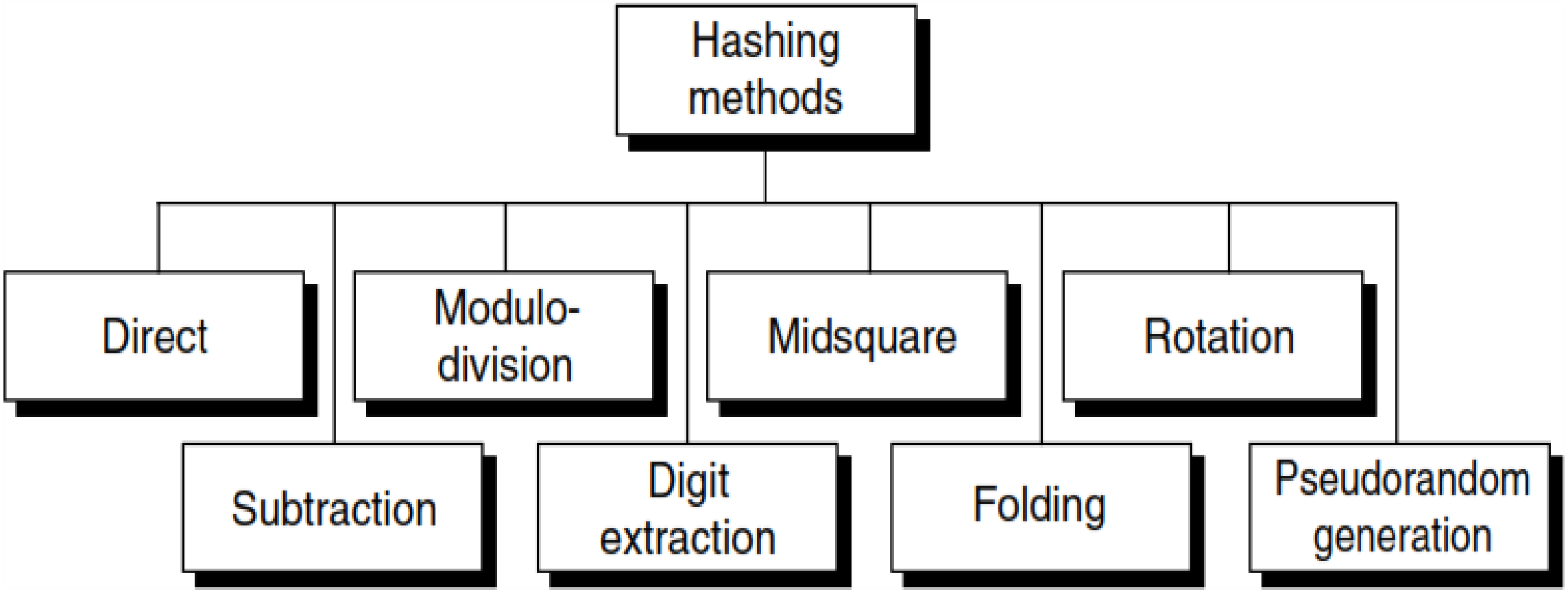
Hash Functions & Collision Resolution Techniques

Dr. S. KAMAKSHI,

AP-III / CSE,

School of Computing

Hashing Methods



Direct Hashing

- Key is used as address without any algorithmic manipulation.
- The data structure must contain an element for every possible key.
- Used very limited, but very powerful because it guarantees that there are no synonyms and therefore no collisions.
- Eg. Applications:
  - Accumulating total sales for each day of a month – A table of size 31 is enough to hold sale amount for each day
  - An organization with 100 employees having employee numbers 1 to 100 – A table of size 100 is enough to hold the employee information

Subtraction Method

- When keys are consecutive but do not start from 1.
- For example, a company having only 100 employees, but the employee numbers start from 1001 to 1100.
- Subtract 1000 from the key to determine the address.
- It is simple and guarantees no collisions.
- Limitations:
  - Can be used only for small lists in which the keys map to a densely filled list

Modulo Division Method

- Also known as division remainder
- Divides the key by the array size and uses the remainder for the address.

  address = key MODULO listSize

- simple hashing algorithm in which listSize is the number of elements in the array
- Works with any list size, but a list size that is a prime number produces fewer collisions than other list sizes.

Digit Extraction Method

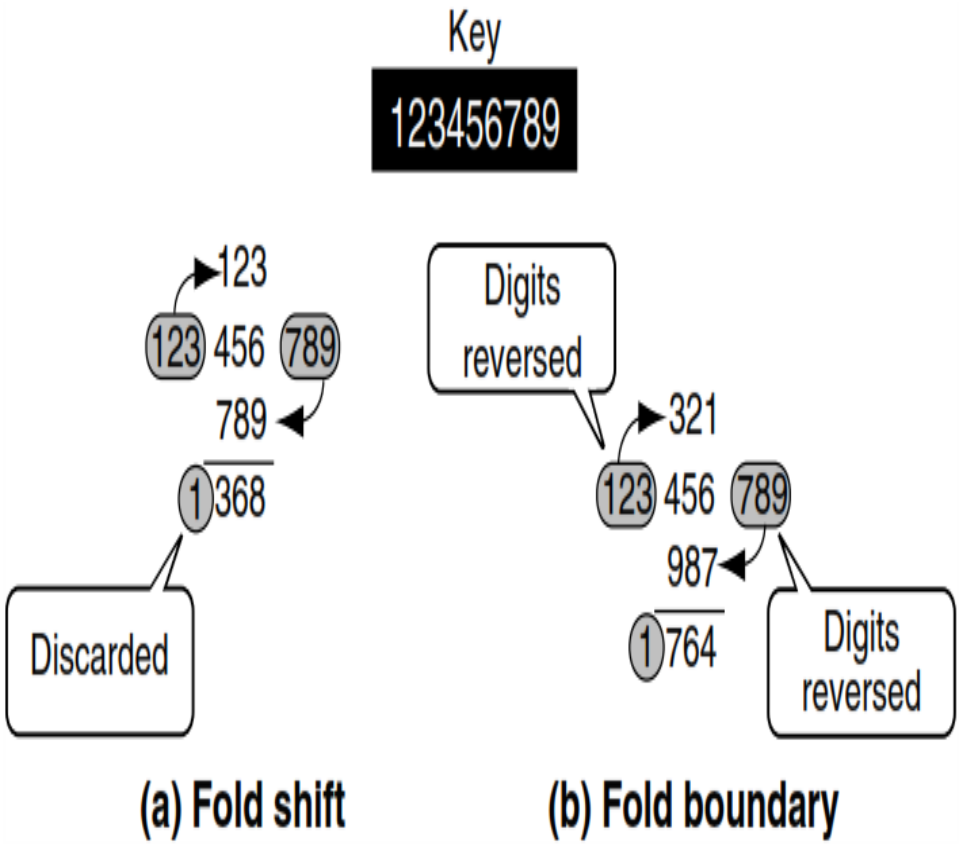
- Selected digits are extracted from the key and used as the address.
- Eg. Selecting 1st, 3rd, 4th digits from a 6-digit key to 3-digit add
  - 379452 ⇨ 394
  - 121267 ⇨ 112
  - 378845 ⇨ 388
  - 160252 ⇨ 102
  - 045128 ⇨ 051

Mid-Square Method

- Key is squared and the address is selected from the middle of the squared number.
- Limitation of this method is the size of the key.
- Eg. For key = 9452,
  - $9452^{2} =$ = 89**3403**04:
  - address is 3403
- Variation on the midsquare method
  - Select a portion of the key, such as the middle three digits, and then use them rather than the whole key.
  - Allows the method to be used when the key is too large to square.
- Eg. 379452: 379^{2} = 143641 ⇨ 364
  - **121**267: 121^{2} = 01**464**1 ⇨ 464
  - **378**845: 378^{2} = 14**288**4 ⇨ 288
  - **160**252: 160^{2} = 02**560**0 ⇨ 560
  - **045**128: 045^{2} = 00**202**5 ⇨ 202

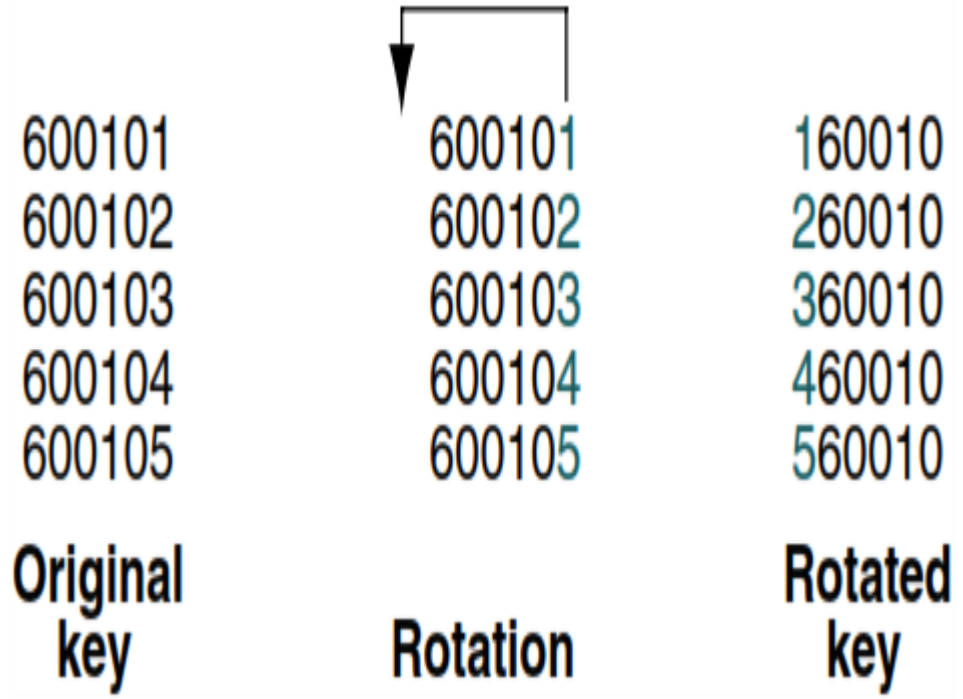
Folding Method

- Two folding methods are used: fold shift and fold boundary.
- In fold shift the key value is divided into parts whose size matches the size of the required address.
- Then the left and right parts are shifted and added with the middle part.
- In fold boundary the left and right numbers are folded on a fixed boundary between them and the center number.
- The two outside values are thus reversed
- The two folding methods give different hashed addresses.



Rotation Method

- Generally not used by itself but rather is incorporated in combination with other hashing methods.
- Most useful when keys are assigned serially, such as employee numbers, part numbers, etc.
- Rotating the last character to the front of the key spreads the data more evenly across the address space
- Used in combination with folding and pseudorandom hashing

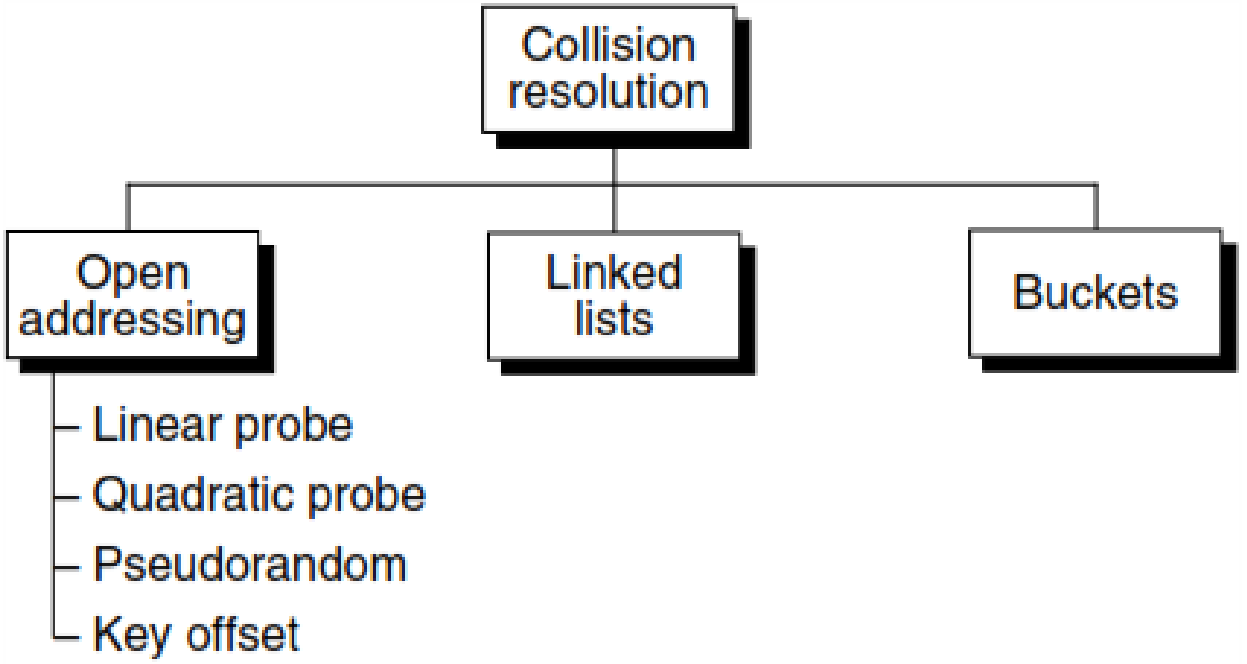


Pseudorandom Hashing

- Key is used as the seed in a pseudorandom-number generator
- The resulting random number is then scaled into the possible address range using modulo-division
- A common random-number generator is
  - y = a x + c
- Eg. a = 17; c = 7; key = 121267
  - y = ((17 * 121267) + 7) modulo 307
  - y = (2061539 + 7) modulo 307
  - y = 2061546 modulo 307
  - y = 41

Collision Resolution

- Collision - when many keys, hash to same address
- All hash functions except direct hashing and subtraction hashing are many-to-one functions: that is, many keys hash to one address.
- How to resolve collision?

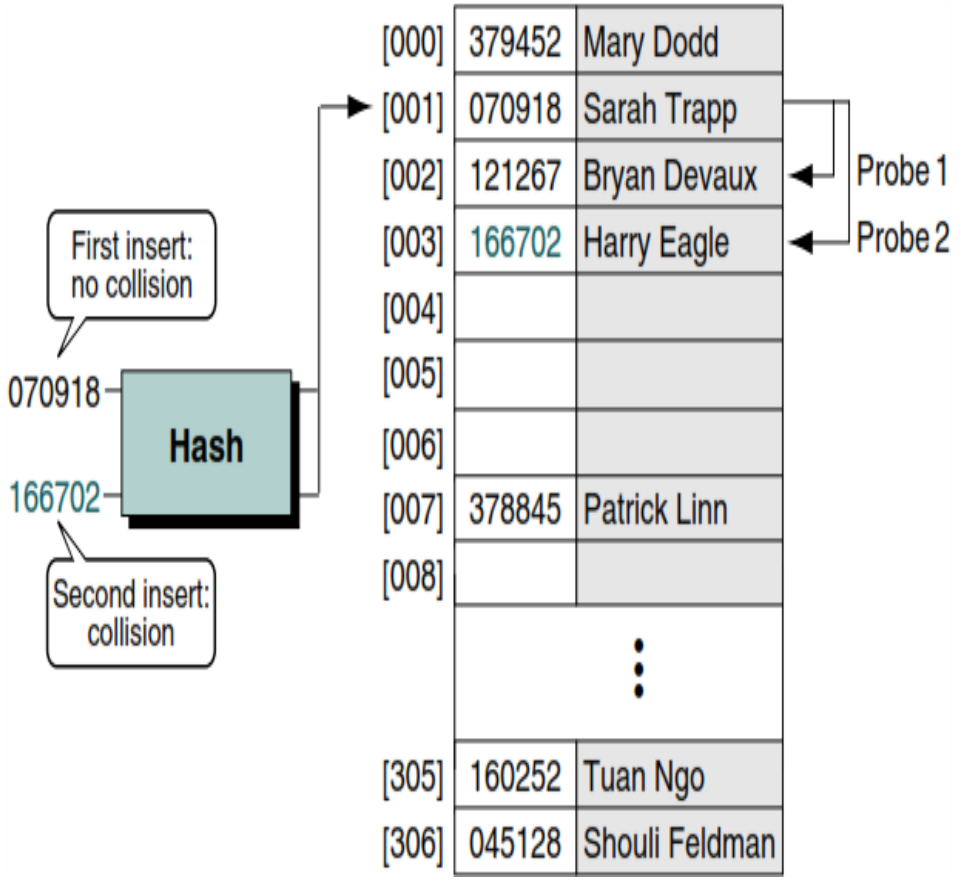


Open Addressing

- Resolves collisions in the prime area—that is, the area that contains all of the home addresses.
- When a collision occurs, the prime area addresses are searched for an open or unoccupied element where the new data can be placed.
- Four different methods:
  - Linear probe
    - Add 1 to the current address until a free space is identified
  - Quadratic probe
    - Increment is the collision probe number squared
  - Double hashing
    - Pseudo random rehashing
      - Hashing again like pseudorandom rehashing method, a random-number generator to rehash the address
    - Key offset hashing
      - Offset is used to rehash the address

Linear Probing

- When data cannot be stored in the home address, resolve the collision by adding 1 to the current address until empty address is found.
- As an alternative to a simple linear probe, add 1, subtract 2, add 3, subtract 4, and so forth until we locate an empty element.
- Eg.: For collision at location 341, check 342, 340, 343, 339, and so forth until an empty address is located.
- If a key hashes to the last location in the list, adding 1 must produce the address of the first element in the list.
- If the key hashes to the first element of the list, subtracting 1 must produce the address of the last element in the list.
- **Advantages:**
  - Quite simple to implement.
  - Data tend to remain near their home address.
- **Disadvantages:**
  - Produce primary clustering.
  - The search algorithm is more complex, especially after data have been deleted.



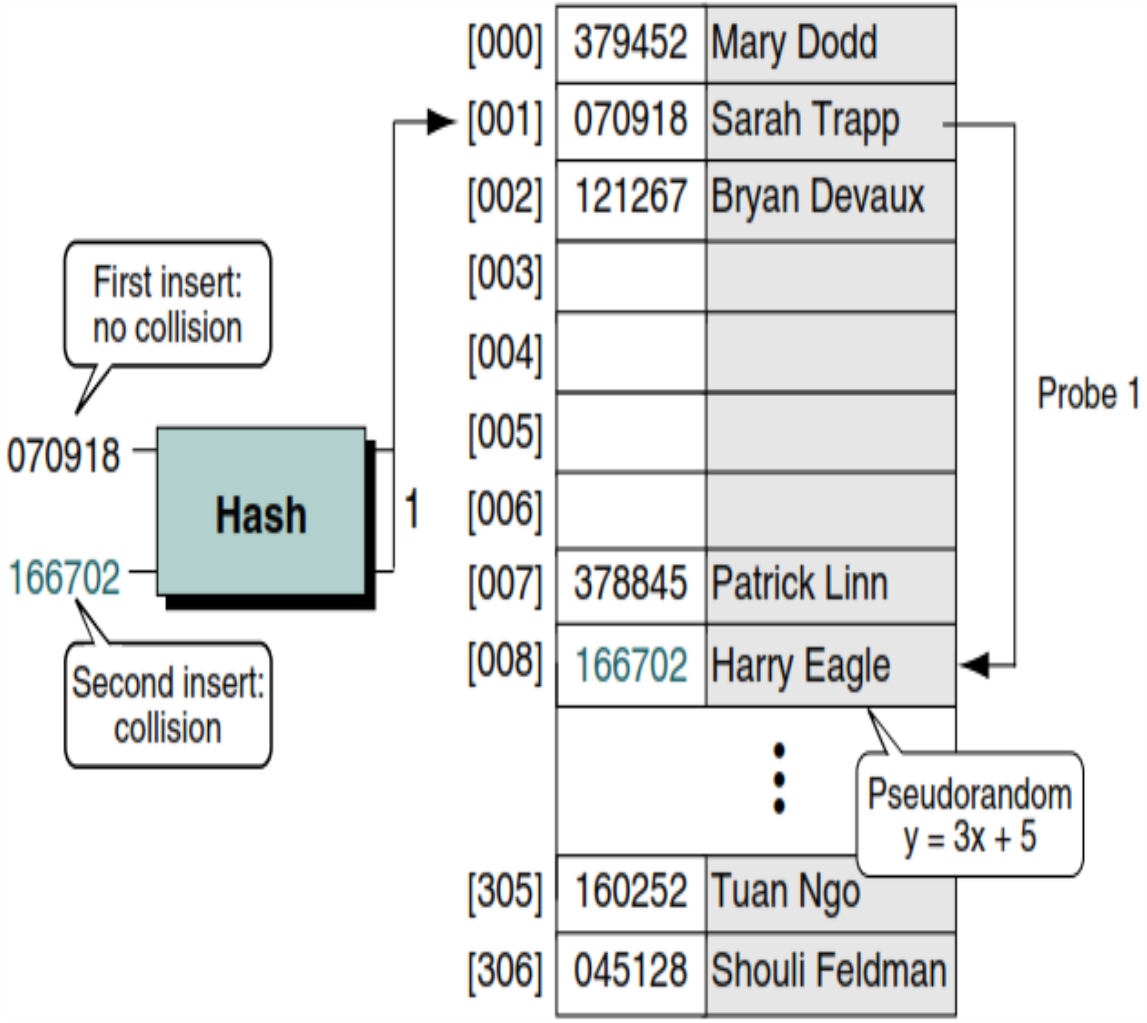
Quadratic Probing

- Primary clustering can be eliminated by adding a value other than 1 to the current address.
- The increment is the collision probe number squared until either find an empty element or exhaust the possible elements.
- **Disadvantages:**
- Time required to square the probe number is high.
- Can be eliminated by using an increment factor that increases by 2 with each probe.
- It is not possible to generate a new address for every element in the list.

Probe number	Collision location	Probe^2 and increment	New address
1	1	$1^2 = 1$	1 + 1 ⇨ 02
2	2	$2^2 = 4$	2 + 4 ⇨ 06
3	6	$3^2 = 9$	6 + 9 ⇨ 15
4	15	$4^2 = 16$	15 + 16 ⇨ 31
5	31	$5^2 = 25$	31 + 25 ⇨ 56
6	56	$6^2 = 36$	56 + 36 ⇨ 92
7	92	$7^2 = 49$	92 + 49 ⇨ 41
8	41	$8^2 = 64$	41 + 64 ⇨ 05
9	5	$9^2 = 81$	5 + 81 ⇨ 86
10	86	$10^2 = 100$	86 + 100 ⇨ 86

Pseudo Random Rehashing

- Address is rehashed using pseudo random number generator
- Prevents primary clustering
- **Disadvantages:**
- All keys follow only one collision resolution path through the list. (This deficiency also occurs in the linear and quadratic probes.)
- Create significant secondary clustering



Key offset

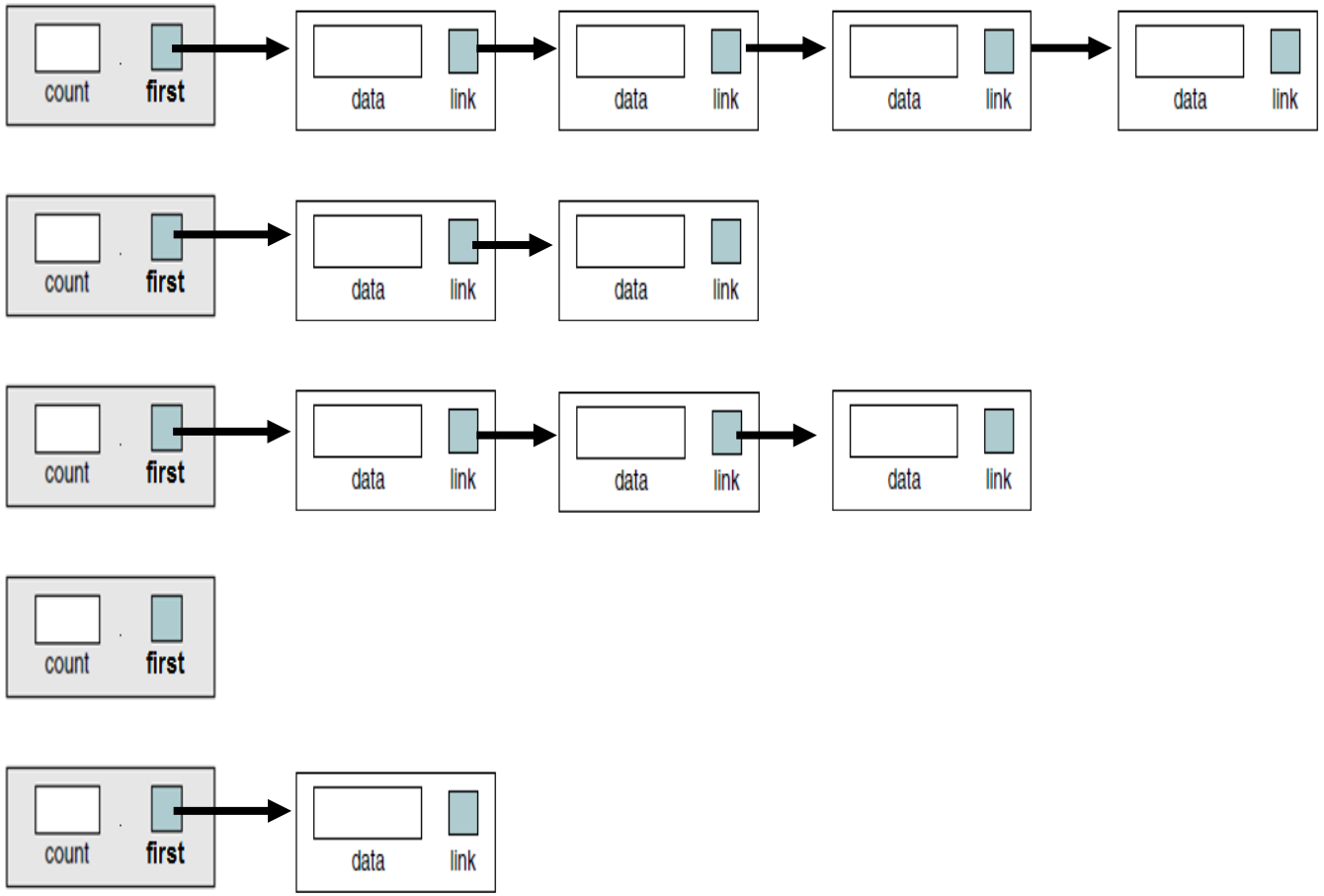
- It is a double hashing method that produces different collision paths for different keys.
- Pseudorandom-number generator produces a new address as a function of the previous address
- Key offset calculates the new address as a function of the old address and the key.
- One of the simplest versions simply adds the quotient of the key divided by the list size to the address to determine the next collision resolution address

$$offSet = \frac{key}{listSize} \qquad address = Remainder\left(\frac{offSet + old\ address}{listSize}\right)$$

Key	Home address	Key offset	Probe 1	Probe 2
166702	1	543	237	166
572556	1	1865	024	047
067234	1	219	220	132

Linked List Collision Resolution

- Major disadvantage of open addressing:
  - Each collision resolution increases the probability of future collisions
- Linked list collision resolution uses a separate area to store collisions and chains all synonyms together in a linked list.
- Uses two storage areas: the prime area and the overflow area.
- Each element in the prime area contains head pointer to a linked list of data that maps to same address.



Bucket Hashing

- Keys are hashed to buckets, nodes that accommodate multiple data occurrences.
- As a bucket can hold multiple data, collisions are postponed until the bucket is full.
- Disadvantages:
  - Uses significantly more space because many of the buckets are empty or partially empty at any given time.
  - It does not completely resolve the collision problem. At some point a collision occurs and needs to be resolved.
  - When it does, a typical approach is to use a linear probe, assuming that the next element has some empty space.