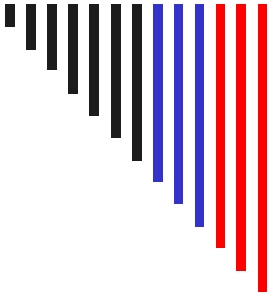


Unit 5

Hashing

Objectives

Upon completion you will be able to

- **Explain Basic Hashing concepts**
- **Methods**
- **Collision resolution**

HASHING Basic Concepts

In a hashed search, the key, through an algorithmic function, determines the location of the data.

Because we are searching an array, we use a hashing algorithm to transform the key into the index that contains the data we need to locate.

Another way to describe hashing is as a key-to-address transformation in which the keys map to addresses in a list.

Hashing is a key-to-address mapping process.

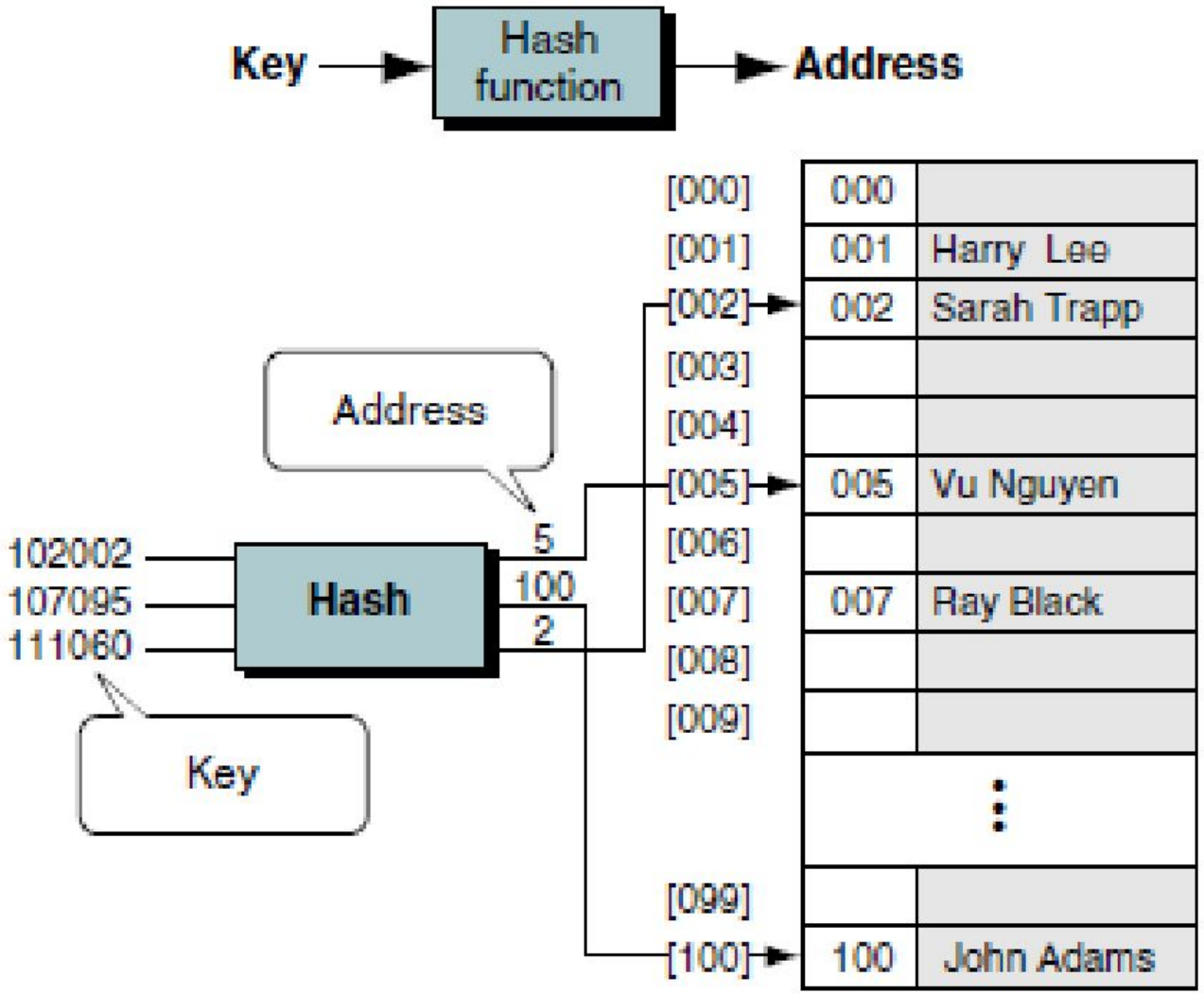


FIGURE 13-6 Hash Concept

Synonyms – The set of keys that hash to the same location

Collision - If the actual data that we insert into our list contain two or more synonyms, we can have collisions. A collision occurs when a hashing algorithm produces an address for an insertion key and that address is already occupied.

The address produced by the hashing algorithm is known as the **home address**.
The memory that contains all of the home addresses is known as the **prime area**.

When two keys collide at a home address, we must resolve the collision by placing one of the keys and its data in another location.

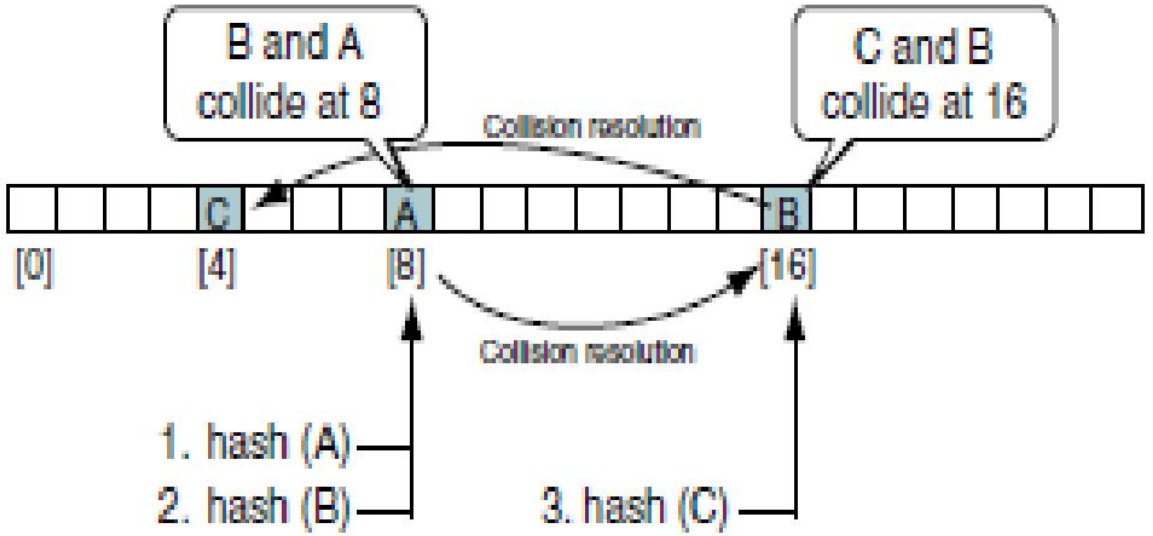


FIGURE 13-7 Collision Resolution Concept

When we need to locate an element in a hashed list, we must use the same algorithm that we used to insert it into the list. Consequently, we first hash the key and check the home address to determine whether it contains the desired element. If it does, the search is complete. If not, we must use the collision resolution algorithm to determine the next location and continue until we find the element or determine that it is not in the list. Each calculation of an address and test for success is known as a **probe**.

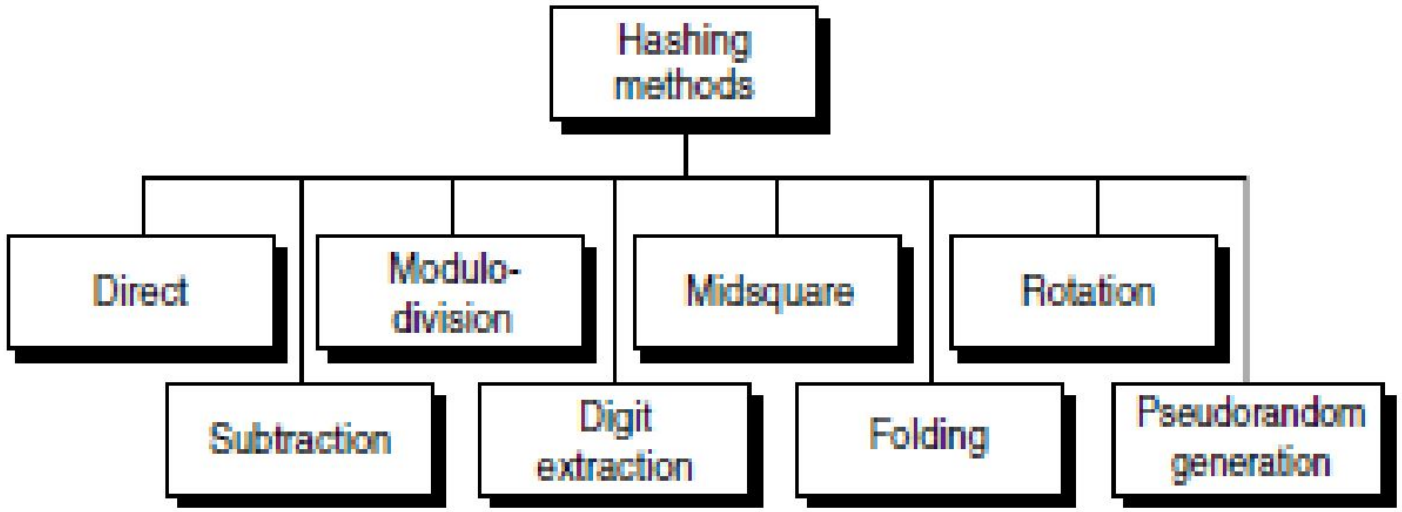


FIGURE 13-8 Basic Hashing Techniques

Modulo-division Method

Also known as division remainder, the modulo-division method divides the key by the array size and uses the remainder for the address. This method gives us the simple hashing algorithm shown below in which `listSize` is the number of elements in the array:

```
address = key MODULO listSize
```

This algorithm works with any list size, but a list size that is a **prime number** produces fewer collisions than other list sizes.

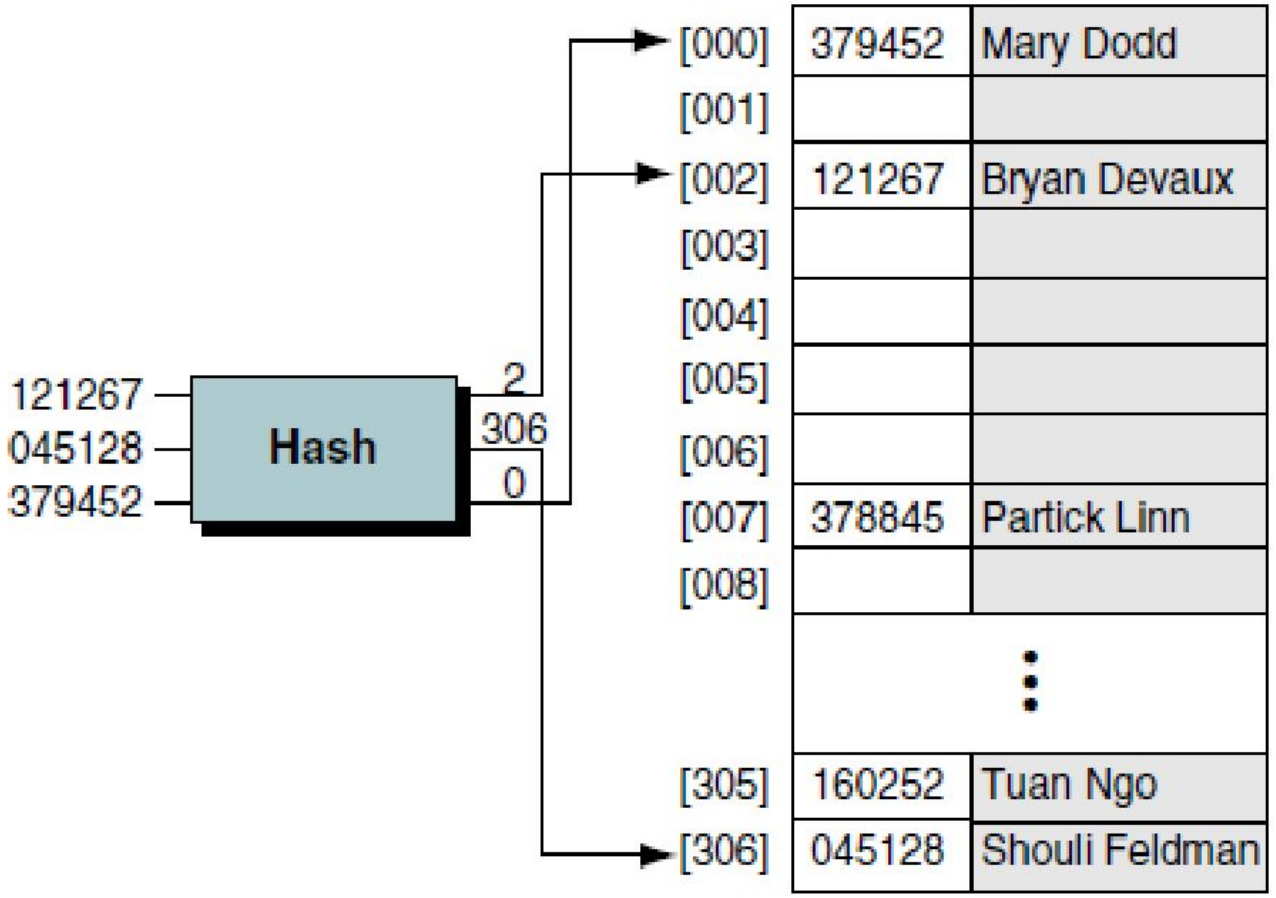


FIGURE 13-10 Modulo-division Hashing

```
121267/307 = 395 with remainder of 2

Therefore: hash(121267) = 2
```

Midsquare Method

In midsquare hashing the key is squared and the address is selected from the middle of the squared number. The most obvious limitation of this method is the size of the key. Given a key of six digits, the product will be 12 digits, which is beyond the maximum integer size of many computers. Because most personal computers can handle a nine-digit integer, let's demonstrate the concept with keys of four digits. Given a key of 9452, the midsquare address calculation is shown below using a four-digit address (0000–9999).

$9452^2 = 89340304$: address is 3403

As a variation on the midsquare method, we can select a portion of the key, such as the middle three digits, and then use them rather than the whole key. Doing so allows the method to be used when the key is too large to square. For example, we can select the first three digits and then use the midsquare method as shown below. (We select the third, fourth, and fifth digits as the address.)

$379452: 379^2 = 143641 \Rightarrow 364$
$121267: 121^2 = 014641 \Rightarrow 464$
$378845: 378^2 = 142884 \Rightarrow 288$
$160252: 160^2 = 025600 \Rightarrow 560$
$045128: 045^2 = 002025 \Rightarrow 202$

Note that in the midsquare method the same digits must be selected from the product.

Folding Methods

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. For example, imagine that we want to map Social Security numbers into three-digit addresses. We divide the nine-digit Social Security number into three three-digit numbers, which are then added. If the resulting sum is greater than 999, we discard the leading digit.

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, where 123 is folded to 321 and 789 is folded to 987. It is interesting to note that the two folding methods give different hashed addresses.

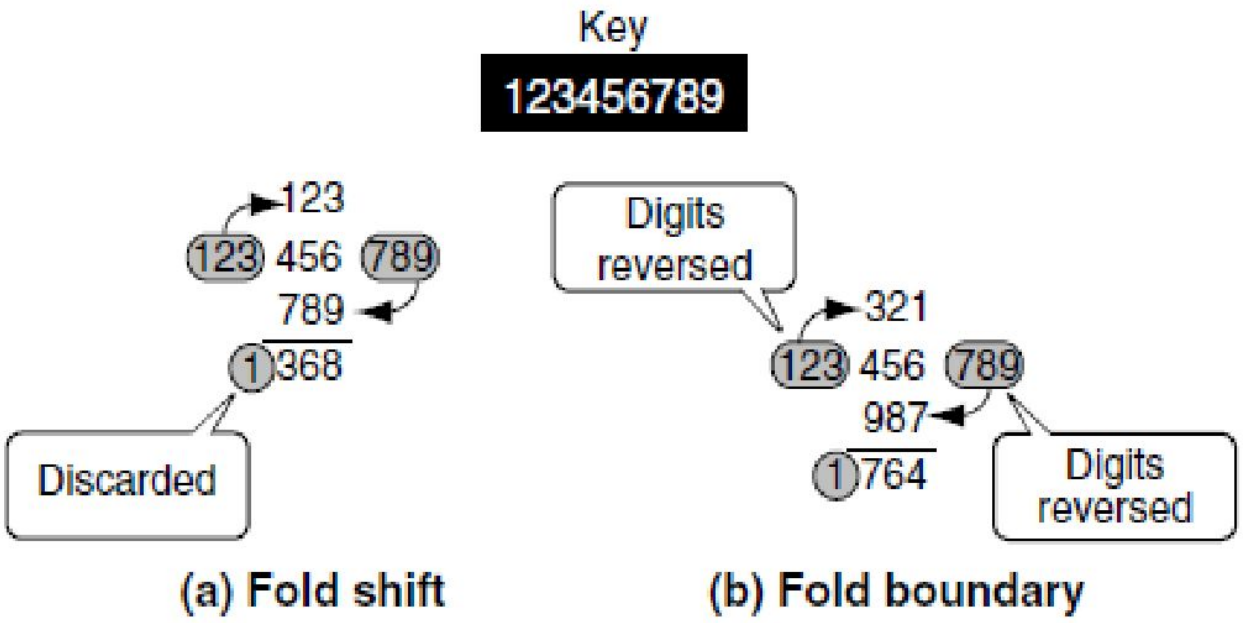


FIGURE 13-11 Hash Fold Examples

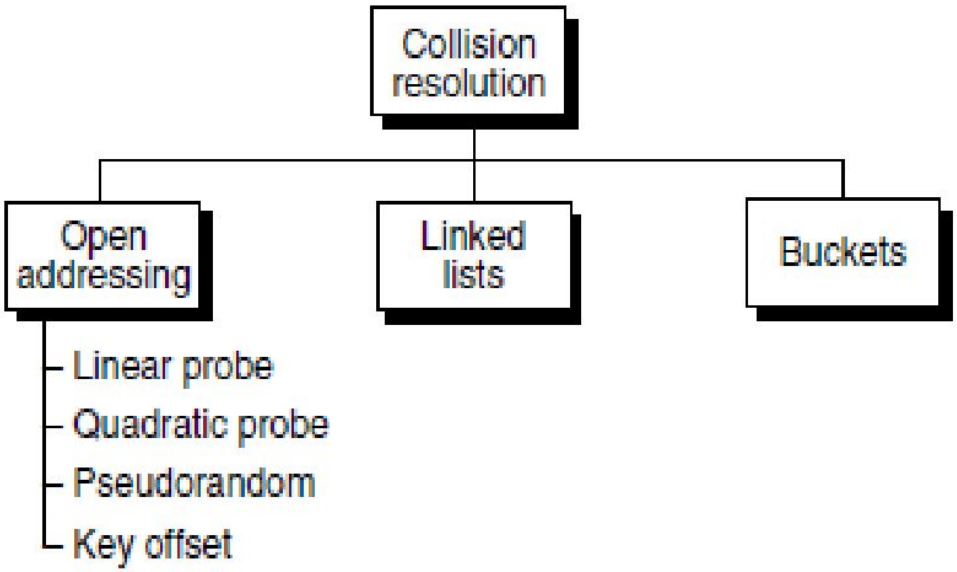


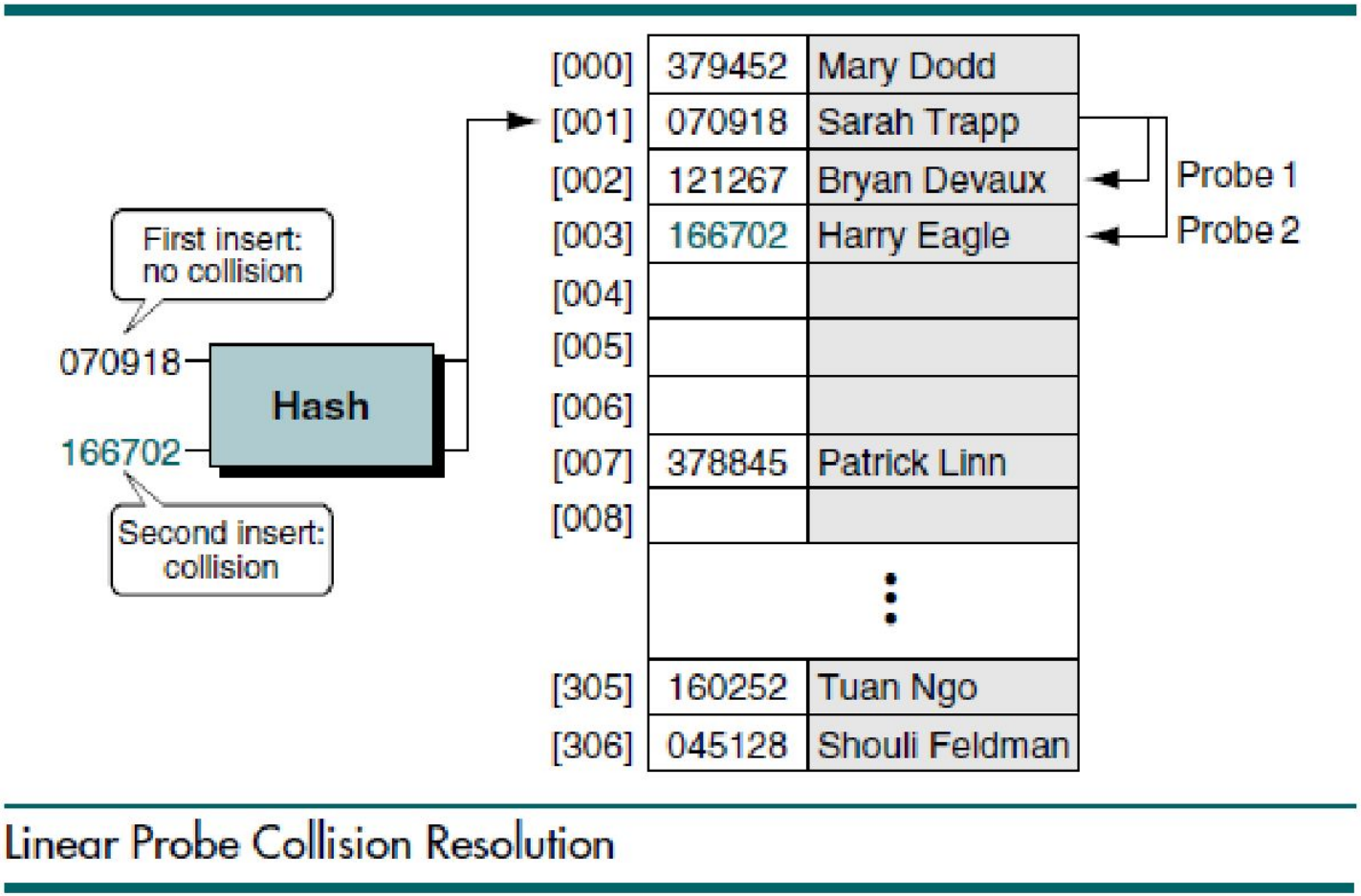
FIGURE 13-13 Collision Resolution Methods

Open addressing resolves collisions in the prime area—that is, the area that contains all of the home addresses.

In Linked list resolution, the collisions are resolved by placing the data in a separate overflow area.

Linear Probe

In a linear probe, which is the simplest, when data cannot be stored in the home address we resolve the collision by adding 1 to the current address.



Linear Probe Collision Resolution

Linear probes have two advantages:

- They are quite simple to implement.
- Data tend to remain near their home address.

Disadvantages:

- Linear probes tend to produce primary clustering(data clustered around home address)
- They tend to make the search algorithm more complex, especially after data have been deleted.

Quadratic Probe

Primary clustering can be eliminated by adding a value other than 1 to the current address. One easily implemented method is to use the quadratic probe. In the quadratic probe, the increment is the collision probe number squared.

To ensure that we don't run off the end of the address list, we use the modulo of the quadratic sum for the new address.

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

Quadratic Collision Resolution Increments

Disadvantages of the quadratic probe:

- Time required to square the probe number
- It is not possible to generate a new address for every element in the list.