FACULTY OF TECHNOLOGY
UNIVERSITY OF COLOMBO
SRI LANKA

Database Management Systems -II

Lecture 12: Hashing in DBMS

Department of Information and Communication Technology

Outline of the syllabus

- Week 1 – Introduction, Revision of Database Design Process and ER
- Week 2 – Enhanced Entity Relationship (EER) -1
- Week 3 – EER Specialization/Generalization Hierarchies, Lattices & Shared Subclasses
- Week 4 – Functional Dependences and Normalization
- Week 5 – Relation Algebra (Revised) and Transact -SQL
- Week 6– Transact - SQL
- Week 7 – SQL Triggers, Functions and Stored Procedures

Outline of the syllabus

- Week 8 – Components of the DBMS – Disks and Files
- Week 9 – Mid Semester Exam
- Week 10 –Indexes, B-Tree and B$^+$-Tree
- Week 11 – B-Tree and B$^+$-Tree (Contd.)
- Week 12 - Physical Database Design and Turning Overview
- Week 13– Transaction and Concurrency Control
- Week 14 – Crash Recovery

In this lecture you will Understand

- What is Hashing in DBMS
- Hashing Technique
- Open Hashing
- Closed Hashing
- What is Hashing in DBMS

Searching Technique

- Linear Search – o(n)

8	5	12	6	15	9	4	3	7	10
0	1	2	3	4	5	6	7	8	9

- Binary Search o(log n)

3	4	5	6	7	8	9	10	12	15
0	1	2	3	4	5	6	7	8	9

- We want a search method with a constant time of search – o(1) .

  This is the requirement to introduce the hashing technique.

Hashing Technique

- List of elements are given, and array space is given to store the element. We can store the elements upon its own index.

- Keys – 8, 3, 7, 6, 4, 9

			3	4		6	7	8	9
0	1	2	3	4	5	6	7	8	9

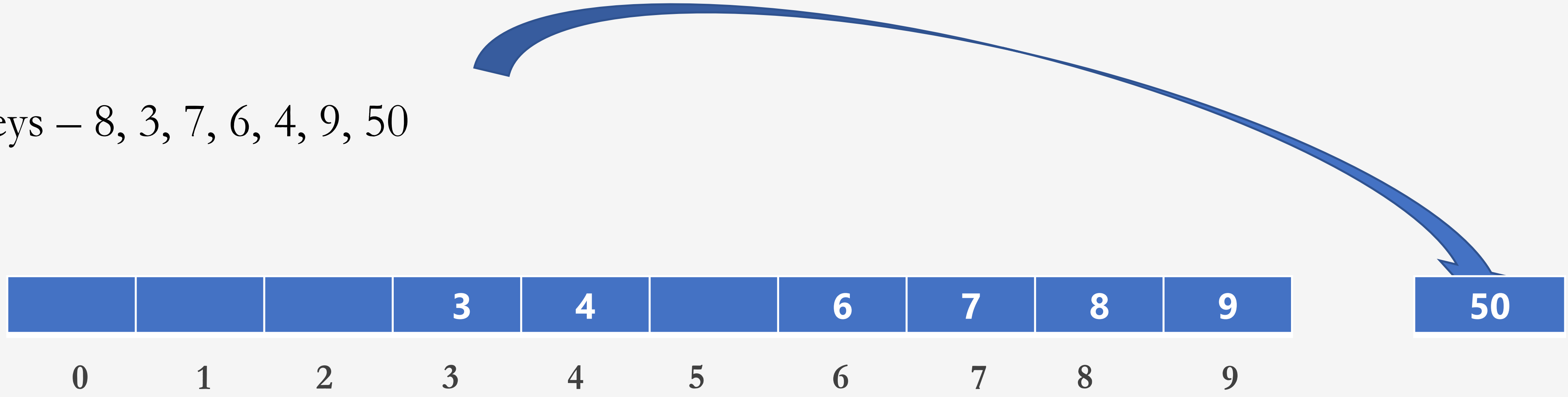
- If we want to search key element 7, it can directly go to the index 7 and see the number

- To search key element 3 it can directly go to the index 3 and read the key element. That is O(1)

Hashing Technique

- if there is a key element 50

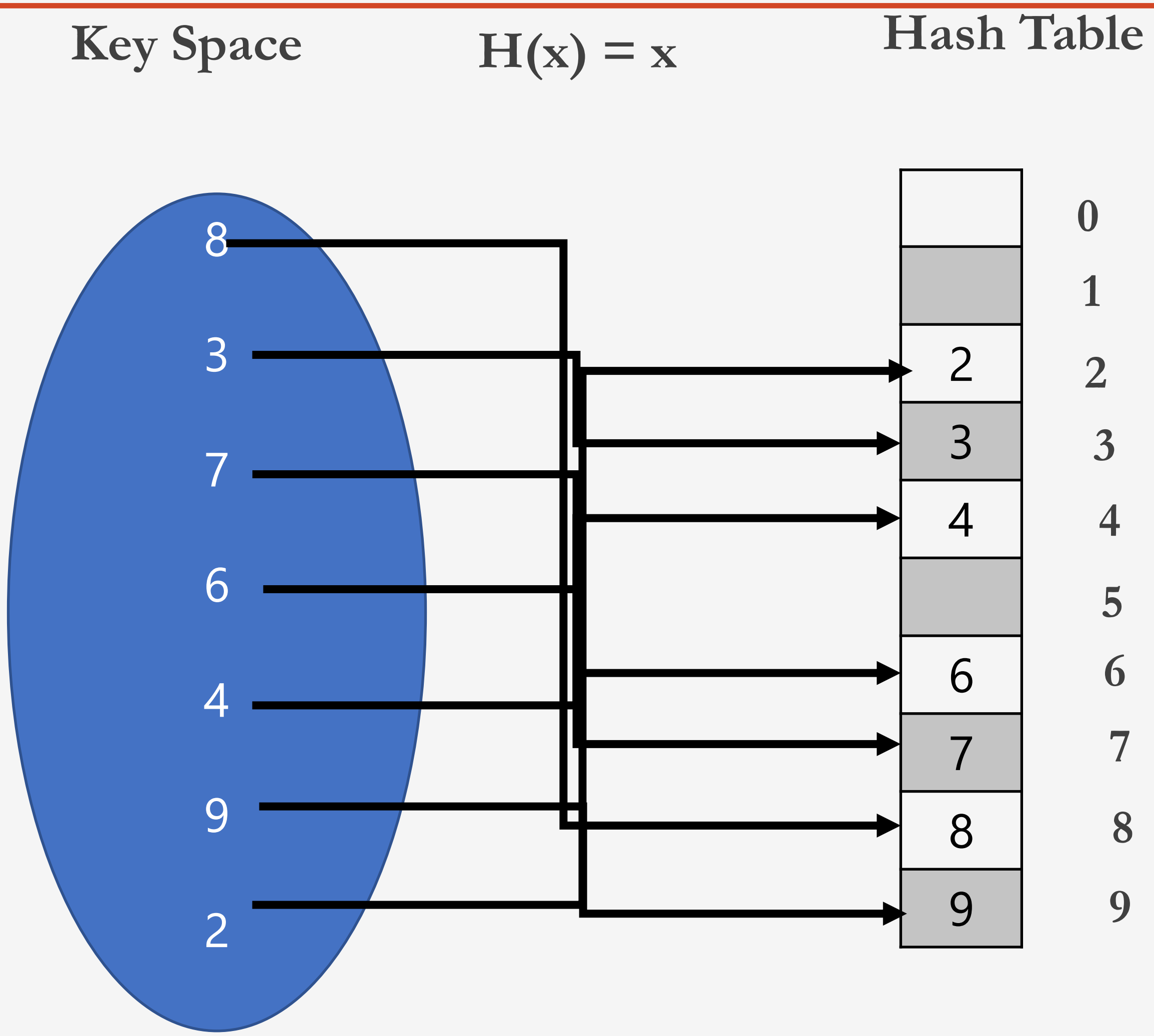
- Keys – 8, 3, 7, 6, 4, 9, 50



			3	4		6	7	8	9	50
0	1	2	3	4	5	6	7	8	9	

- To store the 50 we should have 50 indices in the array. So lot of space is wasted.
- If you want to reduce the time it may waste lot of memories. Just for a new few elements, we need a large array. Most of the spaces are vacant.
- To improve this basic idea, we can use mathematical model for hashing based on functions.

Hashing Technique



- **We can see lot of spaces are in the hash table are vacant**

- Properties of the Functions

  ✓ One-to-one functions

  h(x) = x and function is responsible to mapping the elements in the index to the hash table

  ✓ Many-to-one functions

  ✓ All other functions belong to this.

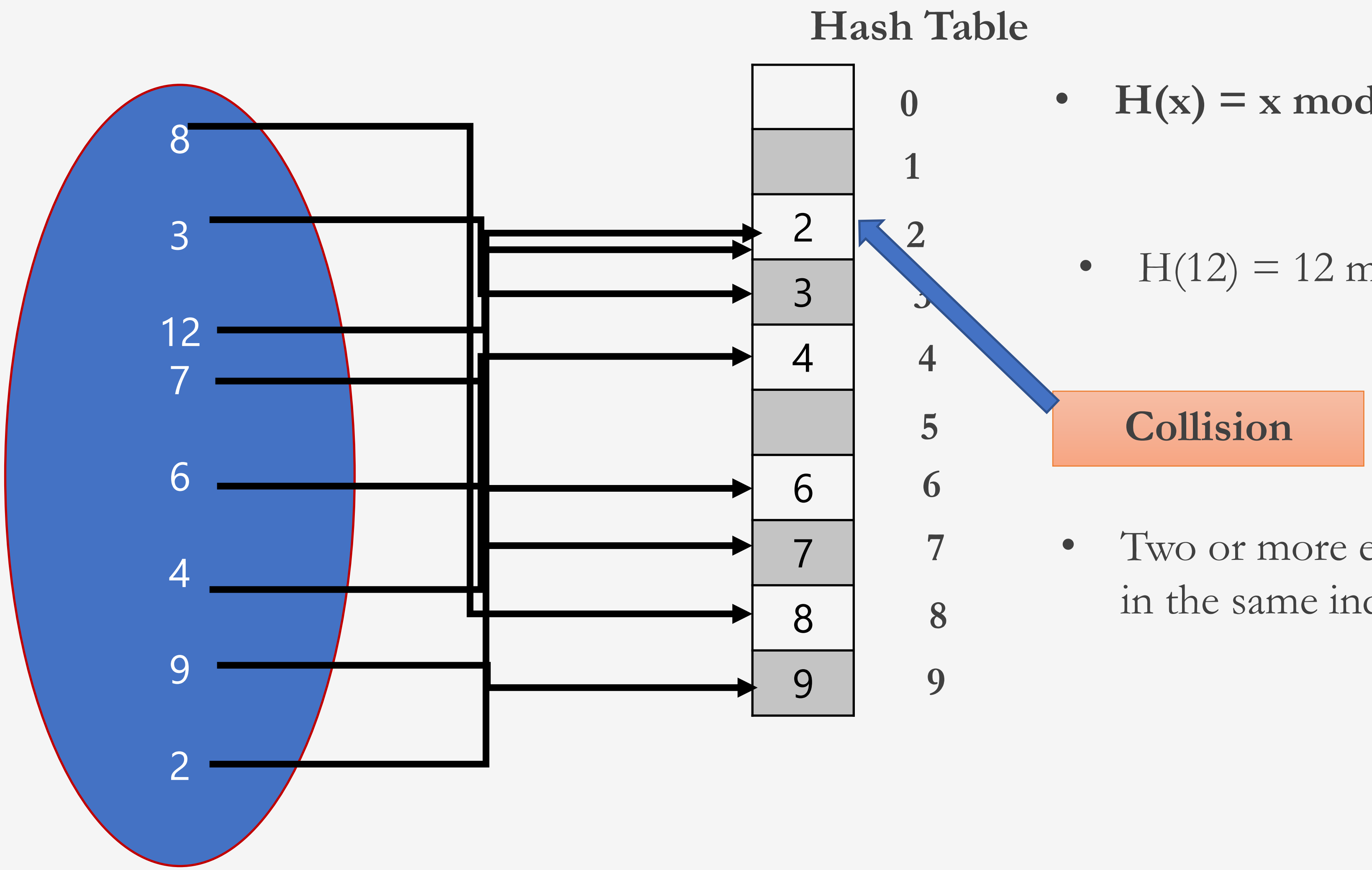
Hashing

- Hashing is a technique that is used to uniquely identify a specific object from a group of similar objects.
- Hash Function : Hash function maps big number or string to a small integer that can be used as index in each table.

Hash Function: h(x) = x Mod m (size of Hash table)

Hashing Technique

Key Space **H(x) = x mod (%) m (size of the array)**



- **H(x) = x mod 10 (reminder)**
- H(12) = 12 mod 10 eminder is 2
- Two or more eliments are matched in the same index.

Collision Resolution Methods

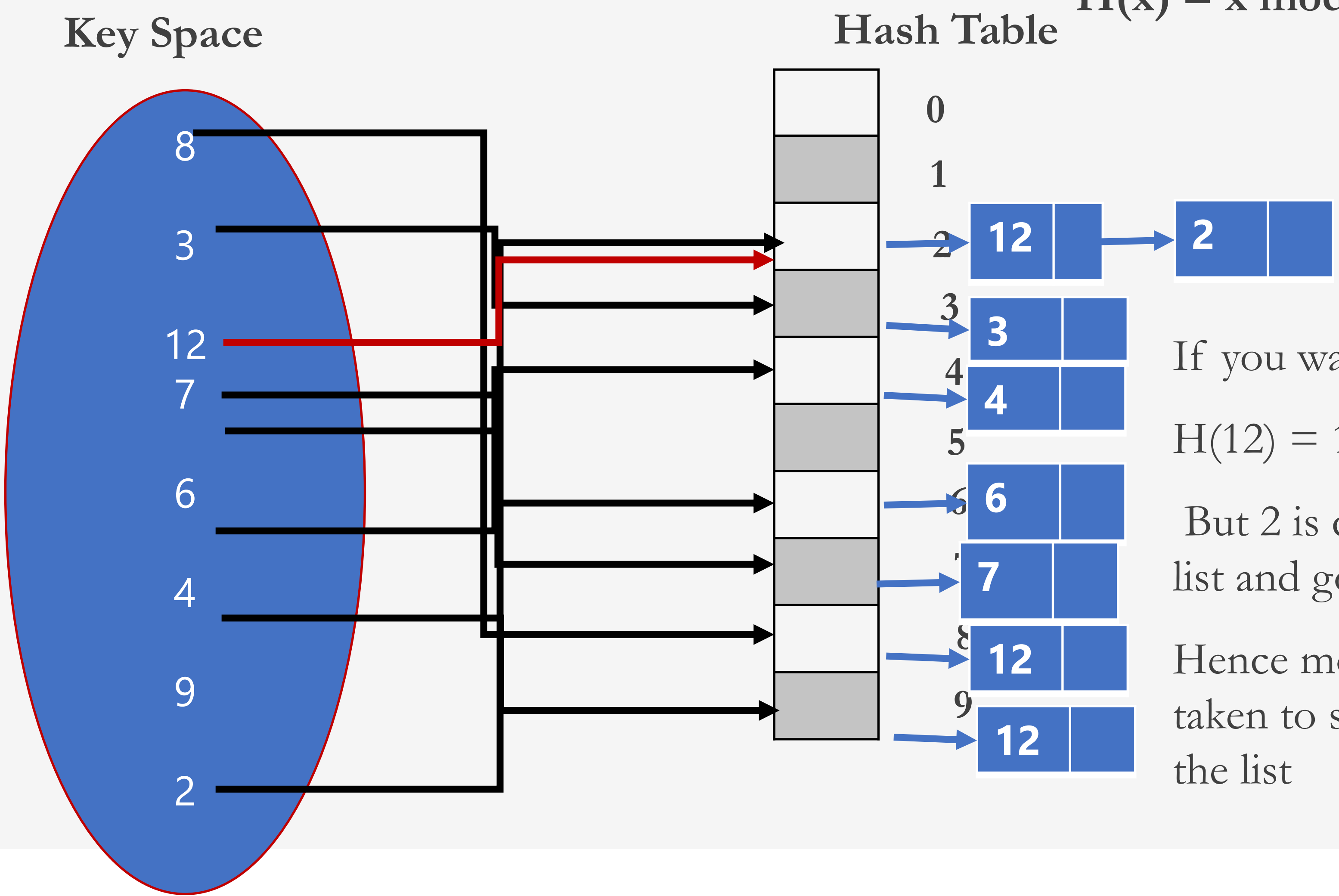
- Open Hashing
  - Chaining

- Closed Hashing
  - Linear Probing
  - Quadratic Probing

Open Hashing

Chaining

H(x) = x mod (%) m (size of the array)



If you want to search key = 12

H(12) = 12 mod 10 = 2

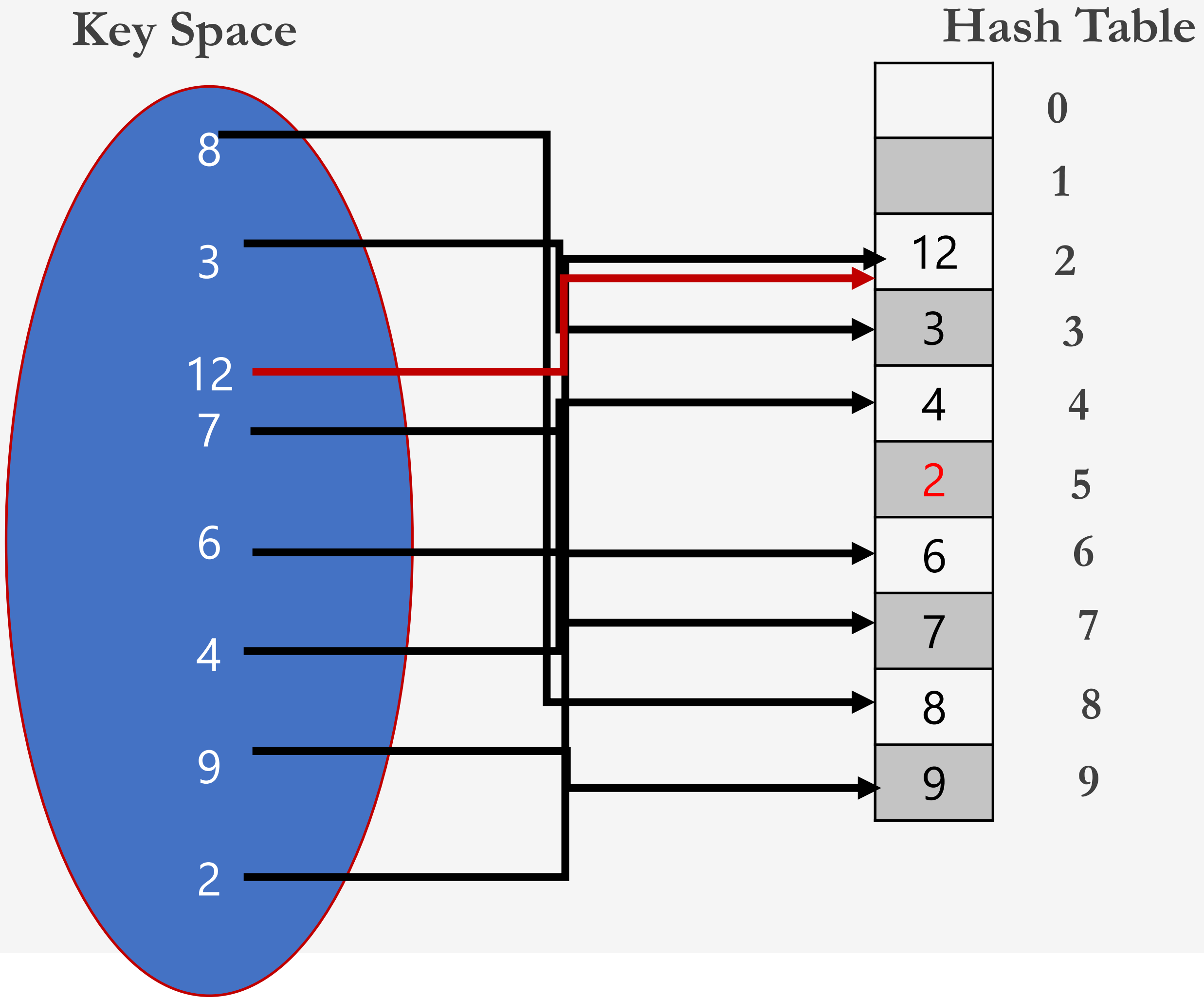
But 2 is connected to 12 from a link list and got the number.

Hence more that o(1) time will be taken to search the key element from the list

Closed Hashing

Linear Probing

H(x) = x mod (%) m (size of the array)



Define the hash function as

H1(x) = [h(x) + f(i)] mod m

f(i) = i where i = 0, 1, 2, … n

In our example 2

H(2) = 2 mod 10 = 2

Key value 12 is there in the 2, then use h(2) = [h(2) + f(0)] i.e is 2

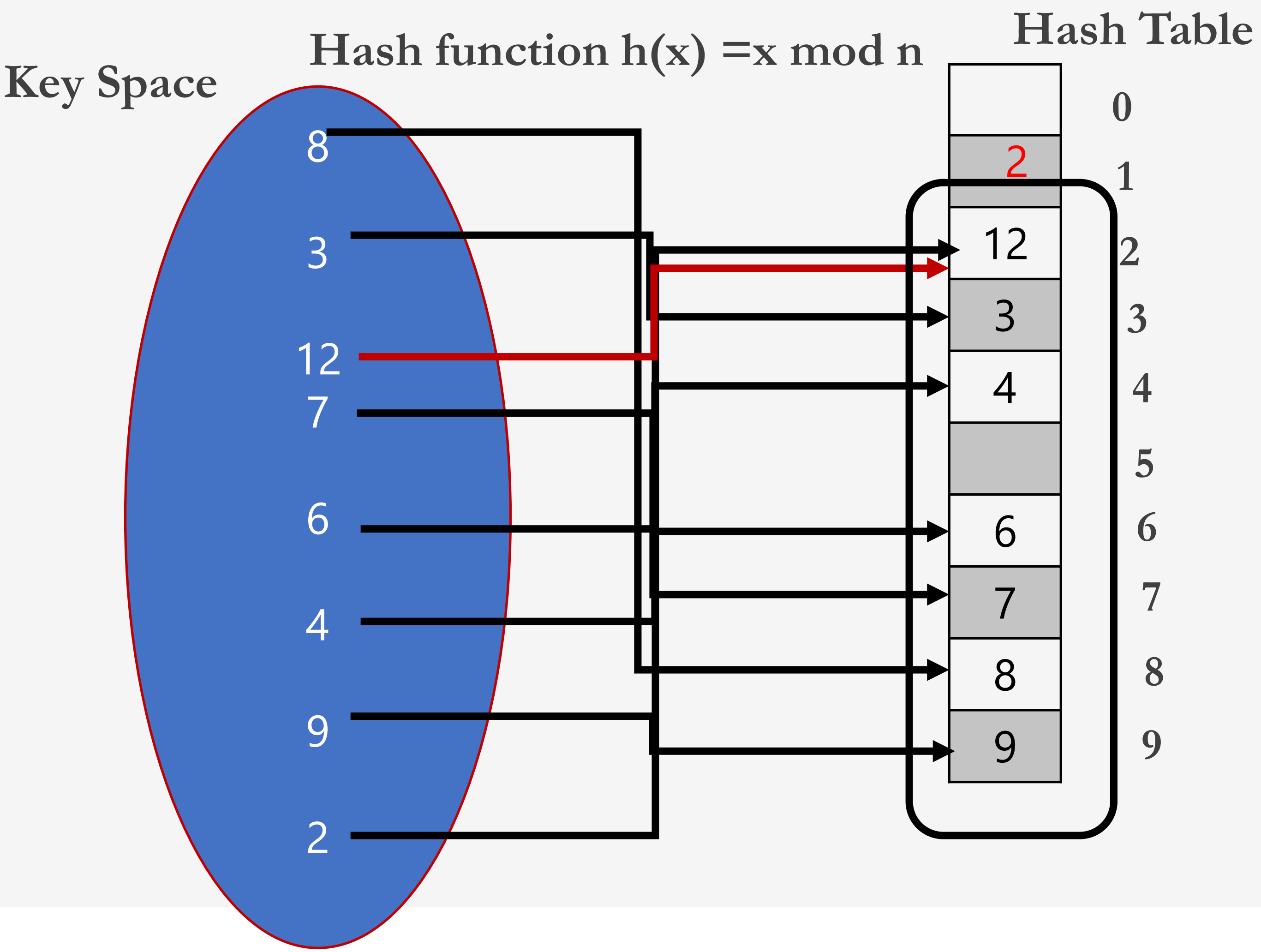
Key value is not there the we have to see h(2) = [h(2) + f(1)] i.e is 3.

Key value is not there the we have to see h(2) = [h(2) + f(2)] i.e is 4.

Key value is not there the we have to see h(2) = [h(2) + f(3)] i.e is 5. the key value is found.

Closed Hashing

Quadratic Probing

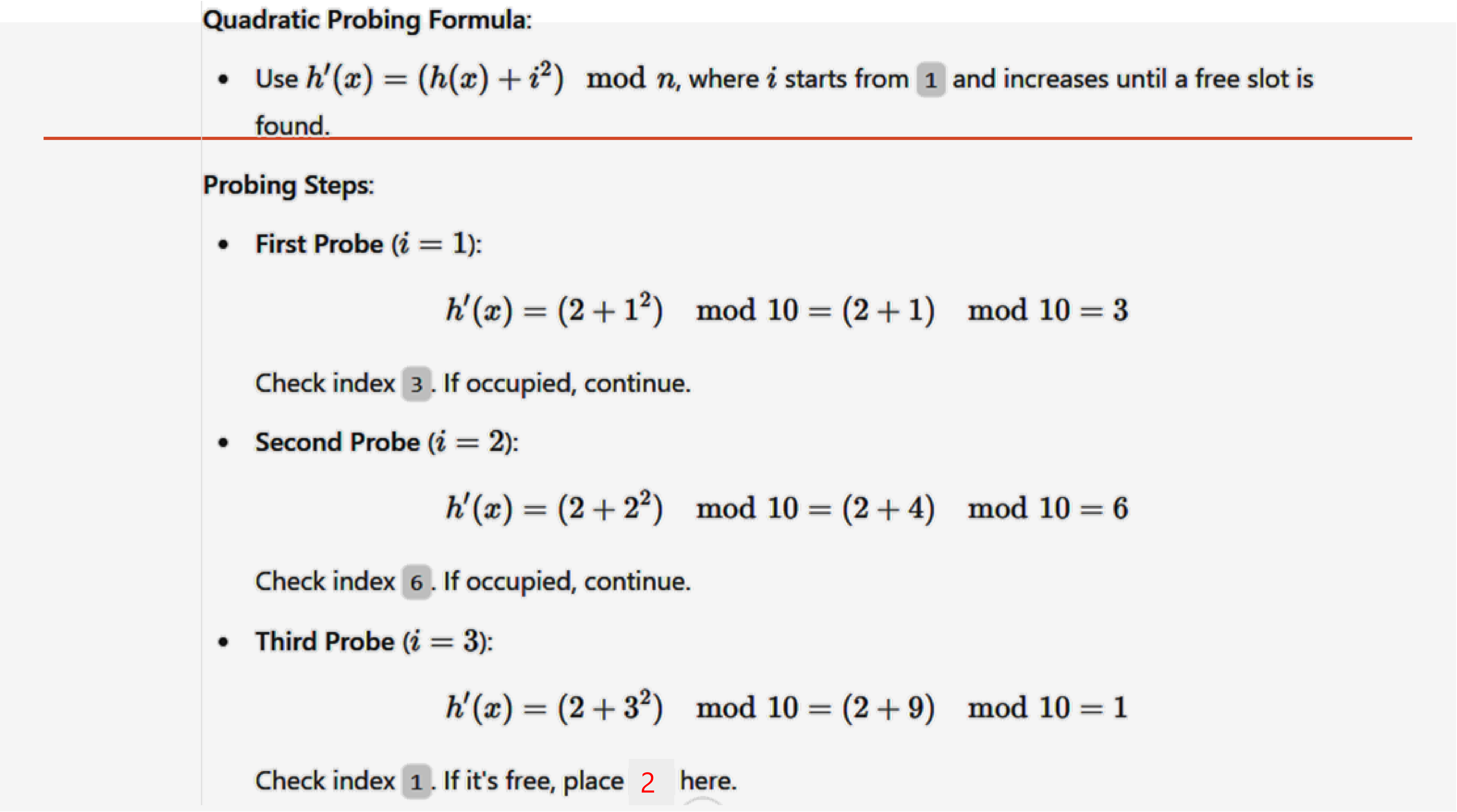


- ✓ Bunch of elements are in a single place at consective locations.
- ✓ This may caouse lot of time to search
- ✓ Avoid the cluster of probing, it can be introduced **Quadratic Probing.**

$h'(x) = [\ h(x) + f(i)]$ mod n (size)

Where $f(i) = i^2$

$i = 0, 1, 2, 3, \ldots.$



Quadratic Probing Formula:

- Use $h'(x) = (h(x) + i^2) \mod n$, where i starts from `1` and increases until a free slot is found.

Probing Steps:

- **First Probe** ($i = 1$):

$$h'(x) = (2 + 1^2) \mod 10 = (2 + 1) \mod 10 = 3$$

Check index `3`. If occupied, continue.

- **Second Probe** ($i = 2$):

$$h'(x) = (2 + 2^2) \mod 10 = (2 + 4) \mod 10 = 6$$

Check index `6`. If occupied, continue.

- **Third Probe** ($i = 3$):

$$h'(x) = (2 + 3^2) \mod 10 = (2 + 9) \mod 10 = 1$$

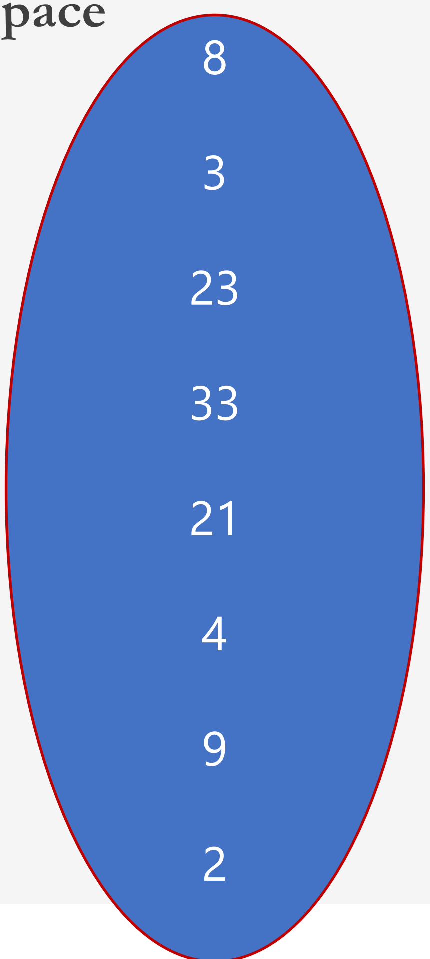
Check index `1`. If it's free, place 2 here.

Closed Hashing

Quadratic Probing

Key Space

Hash function h(x) =x mod n



Hash Table

	0
21	1
2	2
3	3
23	4
4	5
	6
33	7
8	8
9	9

$h'(x) = [\ h(x) + f(i)]$ mod n (size)

Where $f(i) = i^2$

$i = 0, 1, 2, 3, \dots.$

$23 = h'(23) = [(h(23) + f(0)] = 3 + 0 = 3$

$23 = h'(23) = h(23) + f(1^2)] = 3 + 1 = 4$

$33 = h'(33) = [(h(33) + f(0)] = 3 + 0 = 3$

$33 = h'(33) = h(33) + f(1^2)] = 3 + 1 = 4$

$33 = h'(33) = [(h(33) + f(2^2)] = 3 + 4 = 7$

Hashing in DBMS

What is Hashing in DBMS?

- In DBMS, hashing is a technique to directly search the location of desired data on the disk without using index structure.
- Hashing method is used to **index** and **retrieve items** in a database as it is faster to search that specific item using the shorter hashed key instead of using its original value.
- Data is stored in the form of **data blocks** whose address is generated by applying a hash function in the memory location where these records are stored known as a **data block or data bucket**.

Why do we Need Hashing?

- For a huge database structure, it's tough to search all the index values through all its level and then you need to reach the destination data block to get the desired data.
- Hashing method is used to index and retrieve items in a database as it is faster to search that specific item using the shorter hashed key instead of using its original value.
- Hashing is an ideal method to calculate the direct location of a data record on the disk without using index structure.
- It is also a helpful technique for implementing dictionaries.

Important Terminologies Using Hashing?

- **Data bucket** – Data buckets are memory locations where the records are stored. It is also known as Unit Of Storage.
- **Key**: A DBMS key is an attribute or set of an attribute which helps you to identify a row(tuple) in a relation(table). This allows you to find the relationship between two tables.
- **Hash function**: A hash function, is a mapping function which maps all the set of search keys to the address where actual records are placed.
- **Linear Probing** – Linear probing is a fixed interval between probes. In this method, the next available data block is used to enter the new record, instead of overwriting on the older record.
- **Quadratic probing**- It helps you to determine the new bucket address. It helps you to add Interval between probes by adding the consecutive output of quadratic polynomial to starting value given by the original computation.
- **Hash index** – It is an address of the data block. A hash function could be a simple mathematical function to even a complex mathematical function.
- **Double Hashing** –Double hashing is a computer programming method used in hash tables to resolve the issues of has a collision.
- **Bucket Overflow**: The condition of bucket-overflow is called collision. This is a fatal stage for any static has to function.

End of the Lecture – 10-1

Thank You