

DSA through C++

Hashing



Saurabh Shukla (MySirG)

Agenda

- ① why Hashing?
- ② Hashing
- ③ Key terms
- ④ Collision
- ⑤ Hash function
- ⑥ Collision Resolution

Why Hashing

Hashing is designed to solve the problem of needing to efficiently find or store an item in a collection

if we have a list of 1,00,000 words of English and we want to search a given word in the list, it would be inefficient to successively compare the word with all 1,00,000 words until we find a match.

Hashing

It is a technique of mapping keys,
and values into the hash table by
using a hash function.

If implemented correctly, then one
can achieve $O(1)$ time complexity
in searching an item in the
collection of elements.

Efficiency of mapping depends on
the efficiency of the hash function
used.

Key Terms

Hash Table

It is the data structure used to store elements

Hash function It is a function to map (Hash) key-values to the memory address

Hashing

It is a method for storing and retrieving records from a database

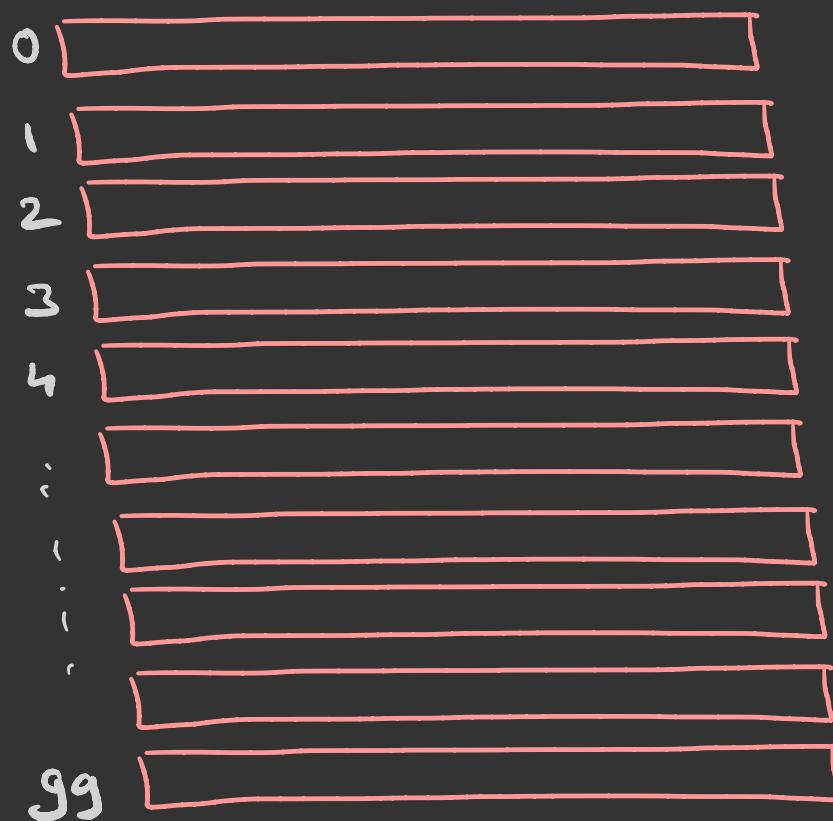
Suppose 100 Student records need to be stored in an array of 100 memory slots. Use hashing to perform efficient access of student data.

Rollno \rightarrow 1 to 100

key

$$i = HF(key)$$

```
int HF ( int key )
{
    return key - 1;
}
```



- Hashing is a method of storing and retrieving records from data structure.
- It lets you insert, delete and search records based on search key value
- When properly implemented these operations can be performed in constant time.
- This is far better than $O(\log_2 n)$ average cost required to do a binary search on a sorted array.

Hashing generally takes records whose key values come from a large range and stores those records in a table with a relatively small number of slots.

RollNo. → 5 digit number

10000 to 99999

10025 → 25
25287 → 87
51128 → 28
48225 → 25

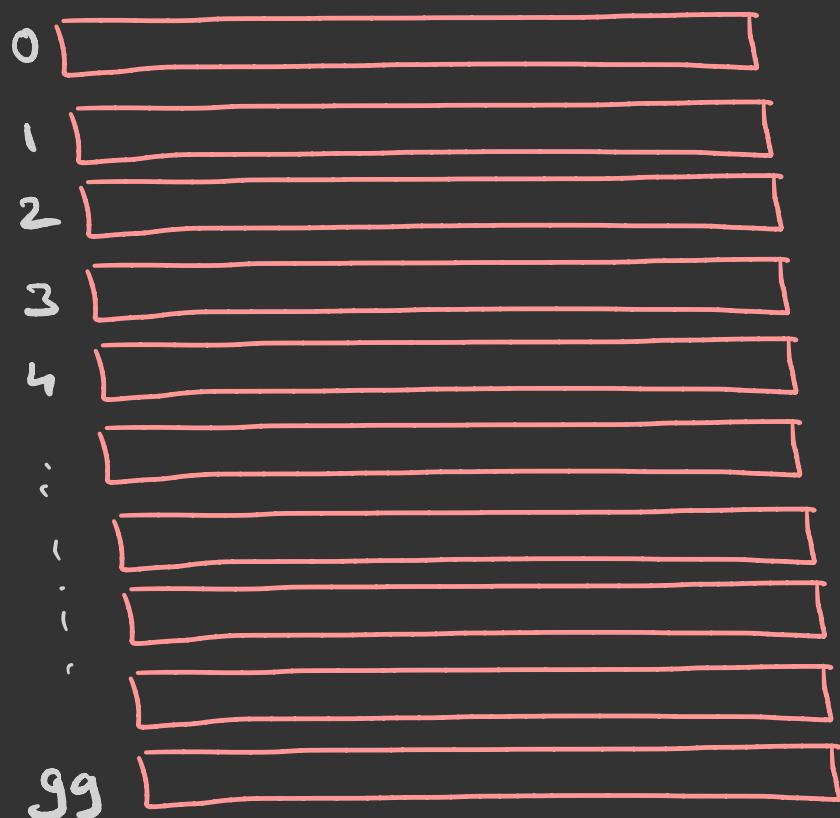
collision

int HF(int key)

{

 return key % 100;

}



Collisions

Collision occur when two records hash to the same slot in the table.

Unfortunately, even under the best of circumstances, collisions are nearly unavoidable.

Hash Function

The hash function creates a mapping between key and value, this is done through the use of mathematical formulas.

Various hash functions

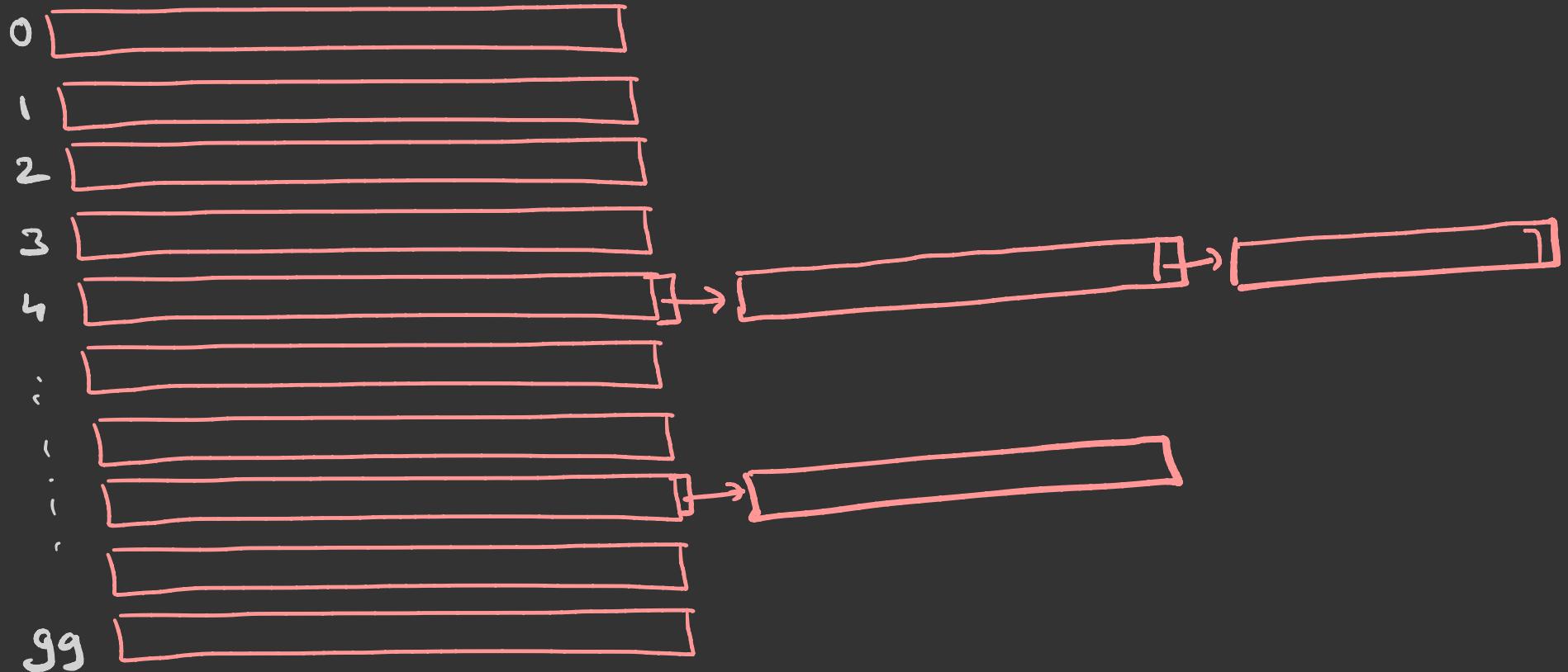
- ① Simple mod function
- ② Division Method
- ③ Binning
- ④ Mid Square method

Collision Resolution

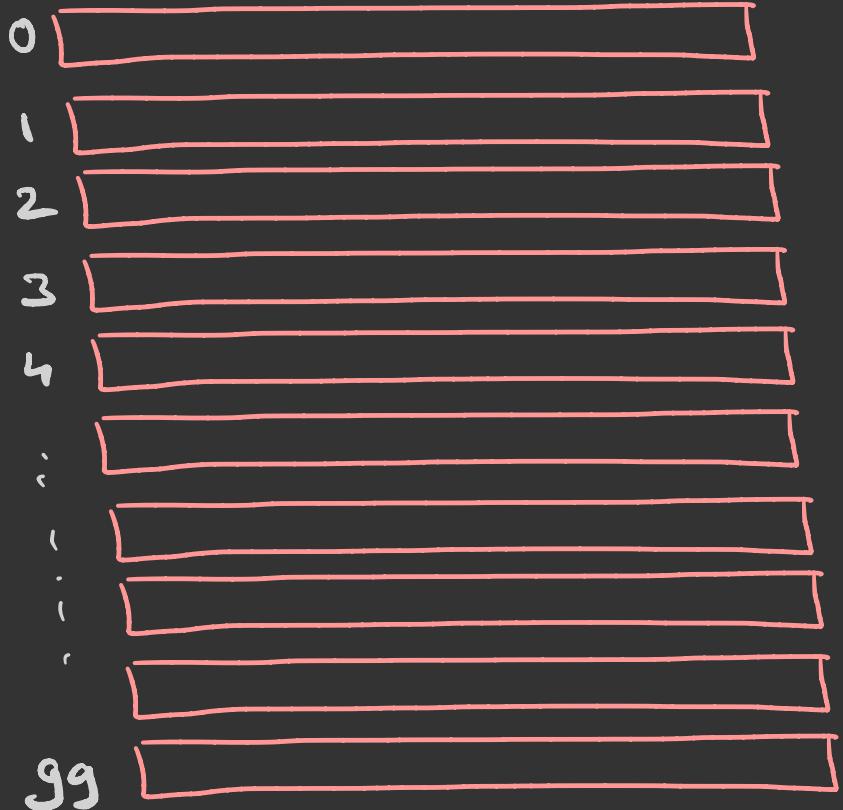
Two types of resolutions:

- ① Open Hashing (chaining)
- ② Closed Hashing (open addressing)

Open Hashing



Closed Hashing



probing

- Linear probing
- quadratic probing
- double hashing