Data Structures and Algorithms- Lab 11

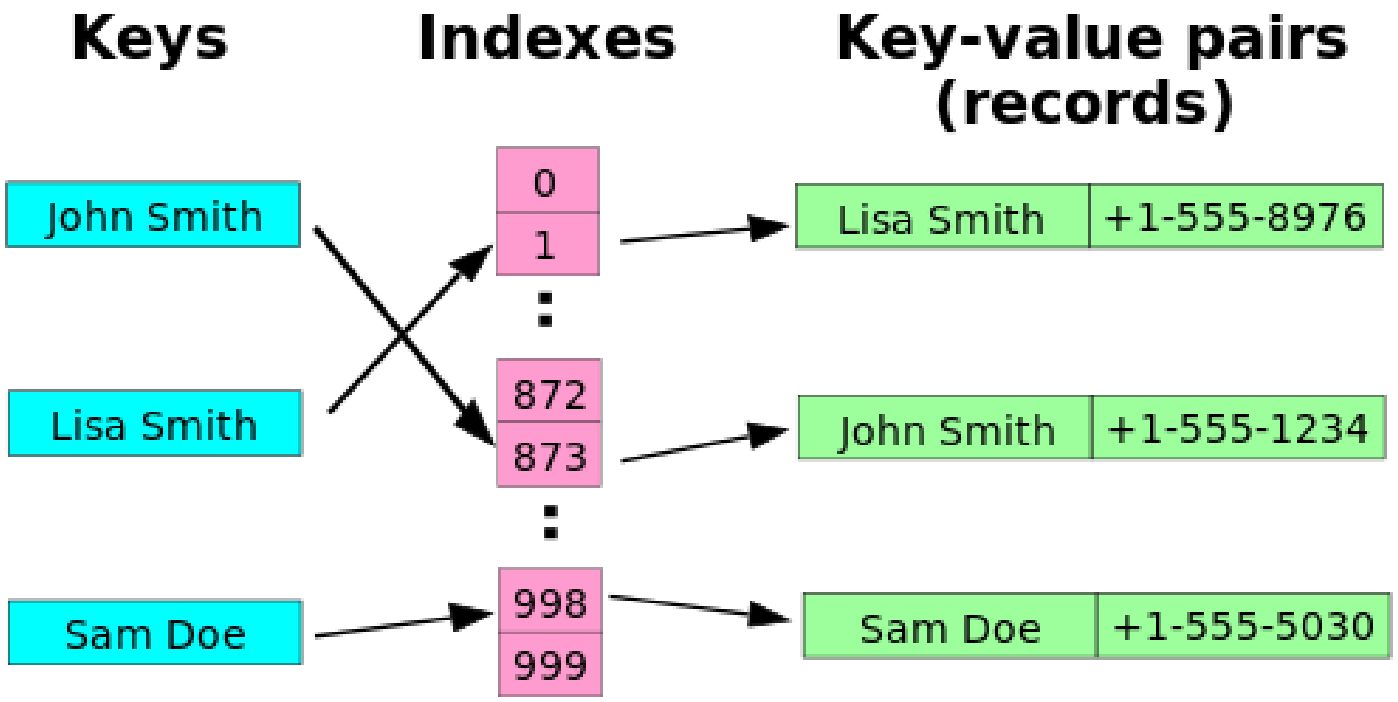
Iulia-Cristina Stanica

Roadmap
Hash Tables

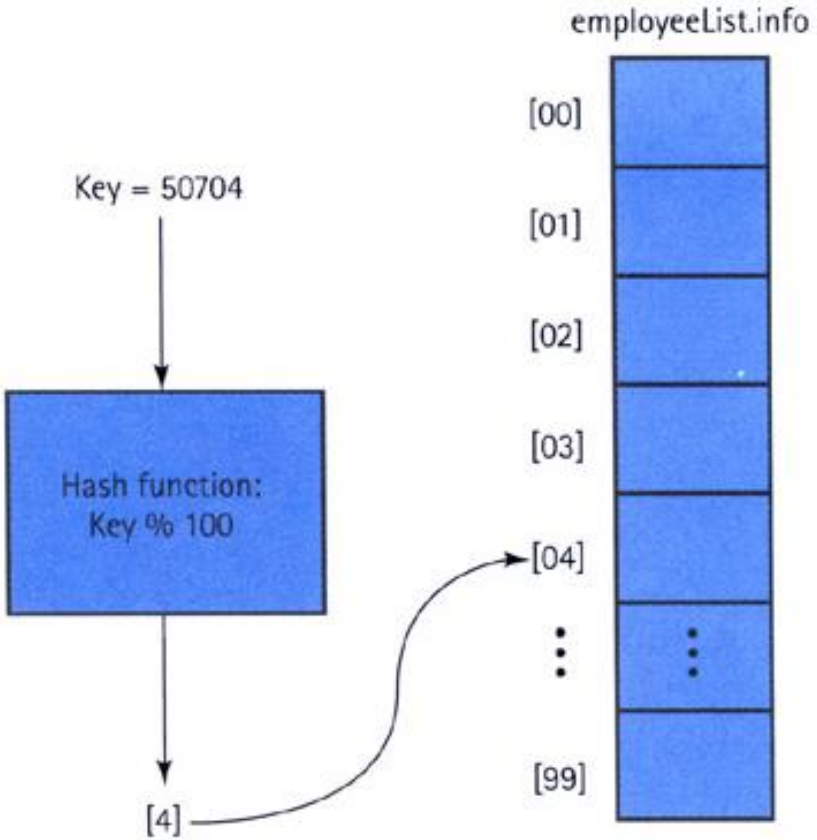
- Hash Tables
- Collision-handling
- C++ implementation

Hash Tables

- **Hash table** = a data structure which allows key – element associations.



Hash function



- Used to identify the place of the element inside the table
- The elements are classified based on a certain function depending on the key: the hash function.

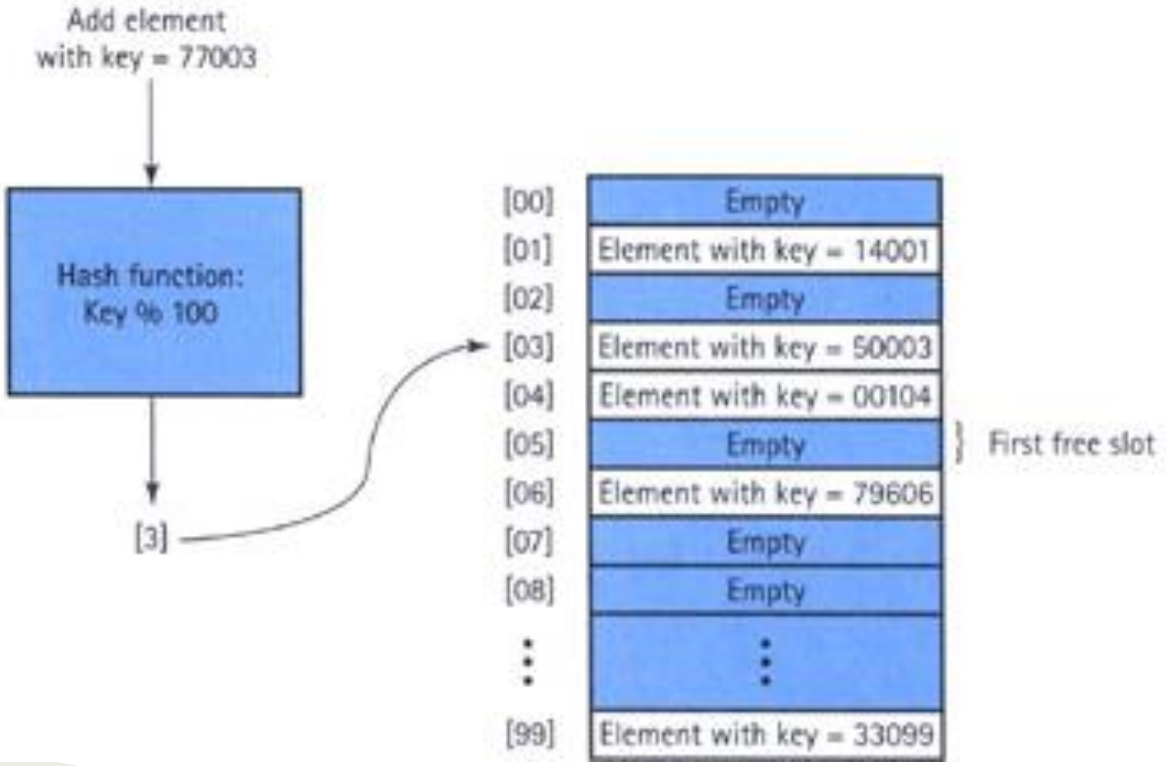
Collisions

- number 01234 and ID number 91234 both "hash" to the same location: list.info [34]
- a good hash function *minimizes collisions* by spreading the elements uniformly throughout the array

Collision-handling algorithms

1. Linear Probing

- store the colliding element in the next available space.

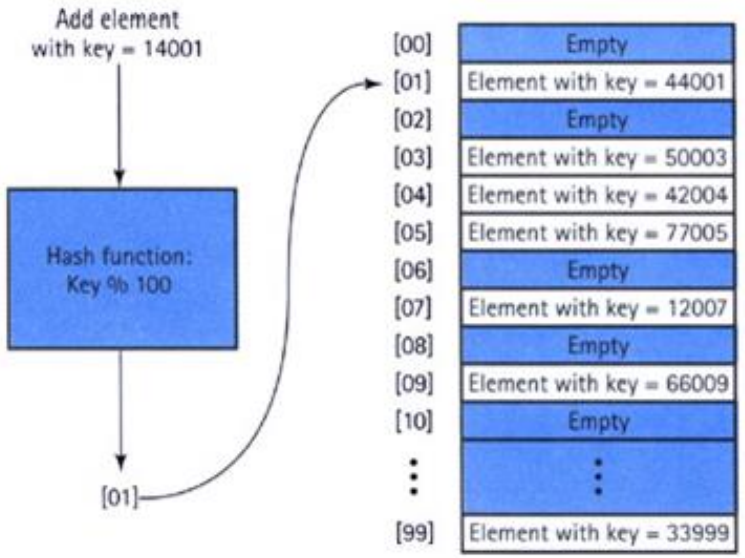


Collision-handling algorithms

2. Rehashing

- Resolving a collision by computing a new hash location from a hash function that manipulates the original location *(HashValueOriginal + constant) % array_size*
  - Obs: the constant and the array-size must be relatively prime

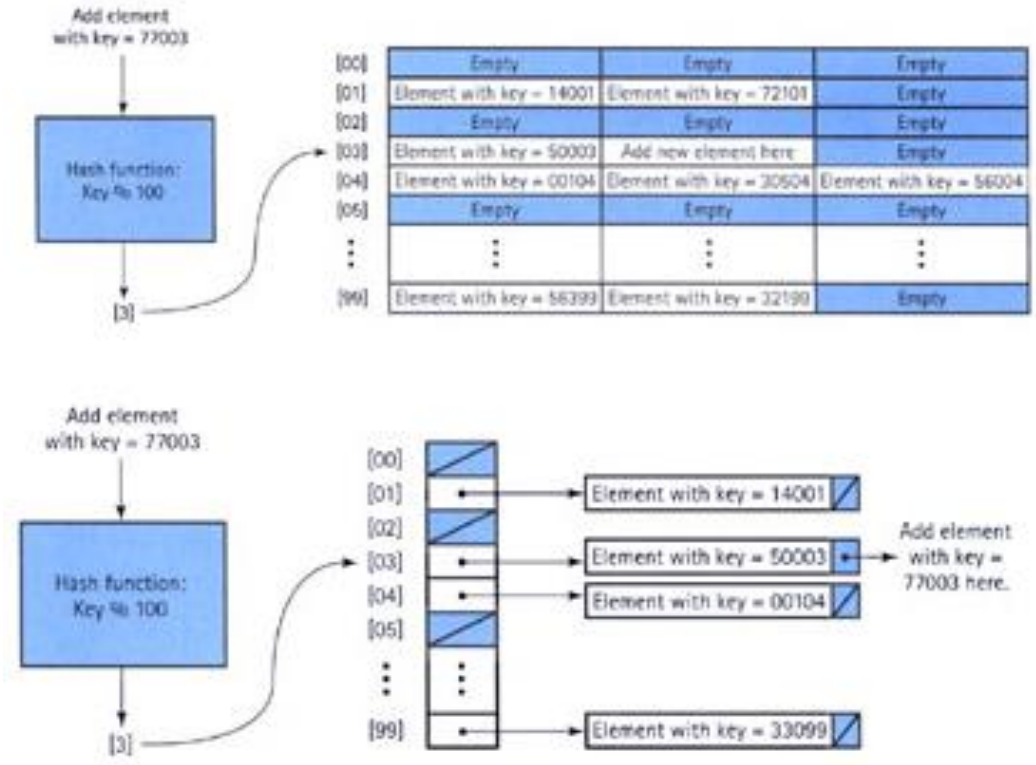
(HashValue + 3) % 100



Collision-handling algorithms

3. Buckets and Chaining

- bucket: a collection of elements associated with a particular hash location
- chain: a linked list of elements that share the same hash location



Exercise 1 (paper / whiteboard)

- Use the following values:
- 113, 117, 97, 100, 114, 108, 116, 105, 99
  Store the values into a hash table with 11 positions.

a) Use the division method of hashing and the linear probing method of resolving collisions.

b) Store the values into a hash table with 5 buckets, each containing 3 slots. If a bucket is full, use the next (sequential) bucket that contains a free slot.

Exercise 2

- Test the header hash.h file in the .cpp file containing the main function by creating an int type hash table. Add items. Test the get and hasKey methods.

- Create a hash table that has keys of type string. Use a hash function created like this:

```
for (int i = 0; i < key.length(); i++)
        hkey = (hkey * P + key[i]) %VMAX;
```

Test the hash table in the main function. Attention! The second parameter in the constructor is the hash function! (we write its name directly)

Exercise 3

- Find a hash function to convert numeric personal numbers (ro. CNP) into values of 4 digits. Test your program.
- Bonus: verify also that the CNP that you read is valid.