UNIVERSITY OF CALIFORNIA
UC RIVERSIDE

Hashing

Chapter 5

Objectives

- Understand the idea of hashing
- Compare hashing to sorting
- Design a hashtable
- Identify the applications that require the hashtable data structure
- Understand the terminology of hashtables
- Distinguish between the different implementations of hash tables

Definition

- hash (verb | \’hæʃ\)
- In Merriam-Webster
  - to chop (food, such as meat and potatoes) into small pieces
  - confuse, muddle

Why Hashing?

- Do we keep everything in an ascending order?

- How do you compare a pair of glasses to a book?

Hashing

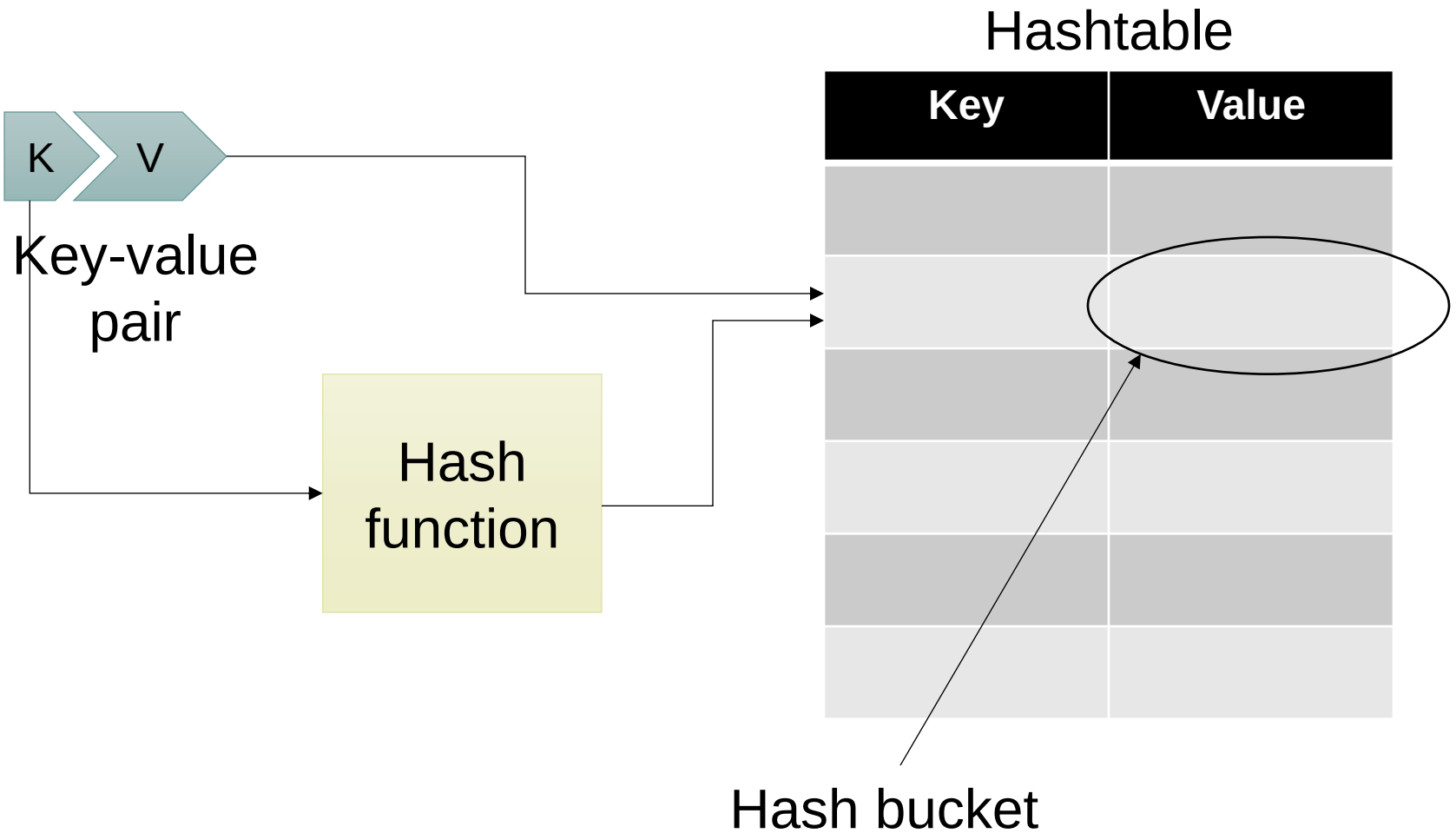
- You store something in a place
- When you want it back, you go and look for it where it is supposed to be
- A simple design: Keep your data elements in a big array of a fixed size so that each element has one fixed position

- What is good/bad about hashing?

Hashtable ADT

- Initialize(n): Initializes an empty hashtable initially with n (empty) slots
- Insert(k, v): Stores the value v with the key k
- Contains?(k): Returns true if there is some value with the key k in the hashtable
- Retrieve(k): Retrieves the value with the key k
- Erase(k): Deletes the value with the key k
- Clear(): Removes all key-value pairs
- Size(): Returns number of elements
- Empty?(): Returns true if the hashtable is empty

Elements of a Hashtable



Design Issues

- What is a good size for a hashtable?
- What are the good and bad properties of a hash function?
  - Fast computation
  - Dispersal (Scatters things around)
  - Memoryless (A must)
- Examples of (bad) hash functions
  - The initial of the last name
  - The student ID modulo number of buckets

A Simple Hashtable

- Key: State names
- Value: Population
- Capacity: 6
- Hash function: Initial letter modulo capacity
- Insert('CA', 40)

Key	Value

A Simple Hashtable

- Key: State names
- Value: Population
- Capacity: 6
- Hash function: Initial letter modulo capacity
- Insert(‘CA’, 40)
- Insert(‘MN’, 5)

Key	Value
CA	40

A Simple Hashtable

- Key: State names
- Value: Population
- Capacity: 6
- Hash function: Initial letter modulo capacity
- Insert(‘CA’, 40)
- Insert(‘MN’, 5)
- Insert(‘NY’, 8)

Key	Value
MN	5
CA	40

A Simple Hashtable

- Key: State names
- Value: Population
- Capacity: 6
- Hash function: Initial letter modulo capacity
- Insert(‘CA’, 40)
- Insert(‘MN’, 5)
- Insert(‘NY’, 8)
- Insert(‘OK’, 4)

Key	Value
MN	5
NY	8
CA	40

Collision

- The biggest problem with hashtables is the collision problem
- Pigeonhole principle
- Birthday paradox
- Hashtables differ mainly on how collisions are handled

Separate Chaining