Hashing

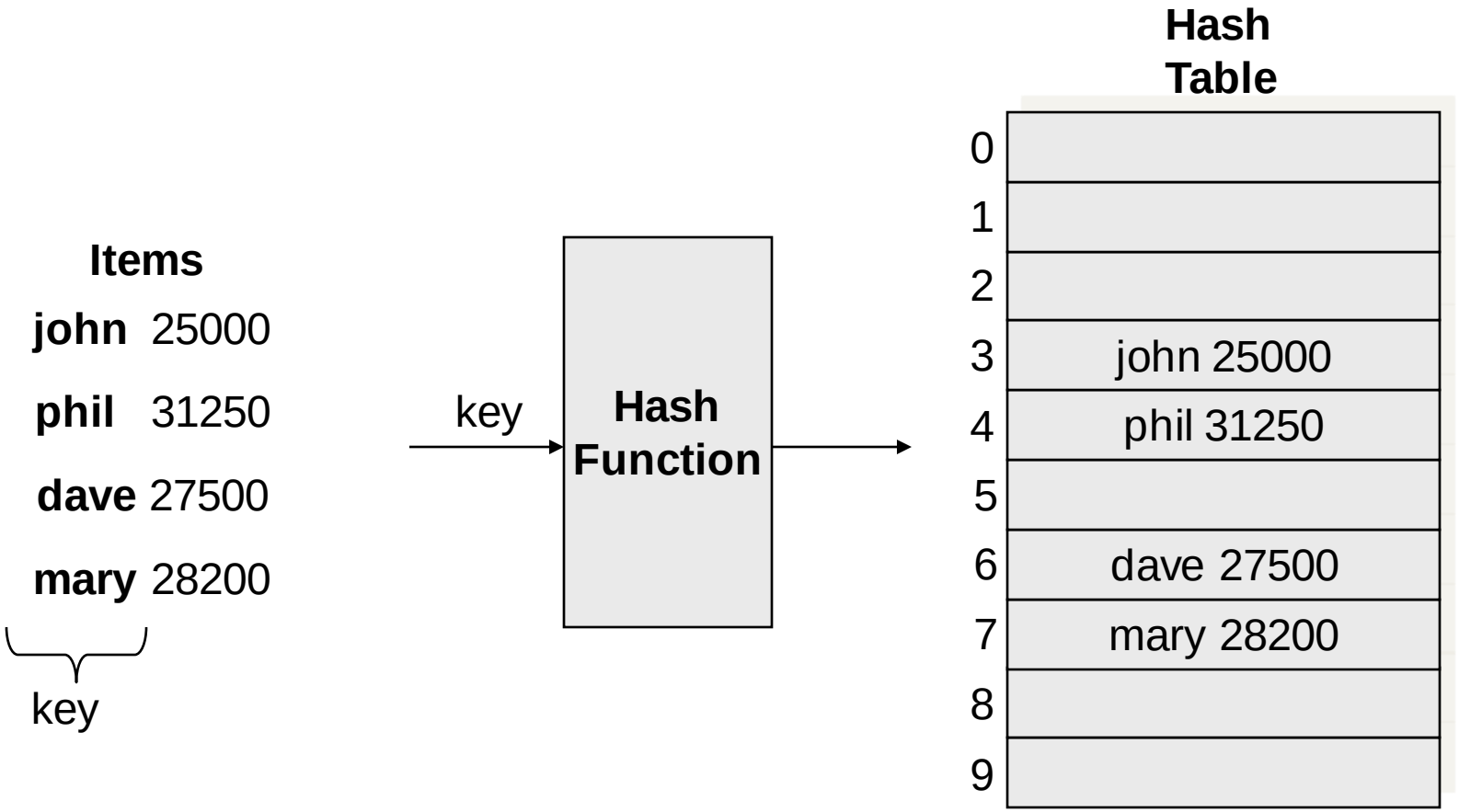
Overview

- Hashing
- Hash function
  - Characteristics of Hash function
- Insert, Update, Delete, and Search operations
- Collision Resolution
  - Separate chaining
  - Open Addressing

Hashing

- Hashing is a technique used for performing the search operation in constant average time (i.e. O(1))
- This data structure, however, is not efficient in operations that require any ordering information among the elements, such as findMin, findMax and printing the entire table in sorted order.

Working Principle



Hash Function

- The hash function:
  - must be simple to compute.
  - must distribute the keys evenly among the cells.
- If we know which keys will occur in advance we can write *perfect* hash functions, but we don't.

Hash function

Problems:

- Keys may not be numeric.
- Number of possible keys is much larger than the space available in table.
- Different keys may map into same location
  - Hash function is not one-to-one => collision.
  - If there are too many collisions, the performance of the hash table will suffer dramatically.

Hash Functions

- If the input keys are integers then simply *Key* mod *TableSize* is a general strategy.
  - Unless key happens to have some undesirable properties. (e.g. all keys end in 0 and we use mod 10)
- If the keys are strings, hash function needs more care.
  - First convert it into a numeric value.

Collision Resolution

- If, when an element is inserted, it hashes to the same value as an already inserted element, then we have a collision and need to resolve it.
- There are several methods for dealing with this:
  - **Separate chaining**
  - **Open addressing**
    - Linear Probing (h(k) +i) mod Size_table
    - Quadratic Probing (h(k) $+c_1 i + c_2 i^2$) mod Size_table
    - Double Hashing

Collision Resolution

- Detailed discussion of code and examples during class.

Summary

- In this lecture
  - The basic concept of hashing is covered along with it's collision resolution techniques.
  - It is concluded that choice of hash function is important to get the constant average time for search operation
  - This data structure is not good for functions like finding max, min, sorting etc.