Fundamentals of
Data Structures using C

Separate Chaining

B.Bhuvaneswaran, AP (SG) / CSE

9791519152
bhuvaneswaran@rajalakshmi.edu.in

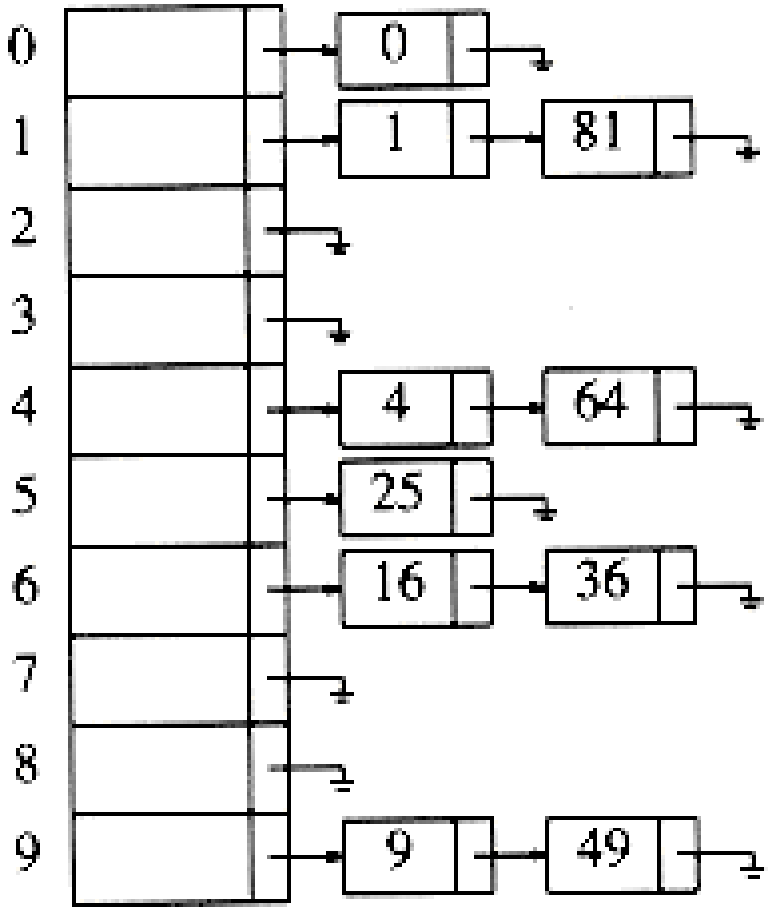
RAJALAKSHMI
ENGINEERING COLLEGE

Introduction

- The separate chaining technique uses linked lists to overcome the problem of collisions.
- In this technique, all the keys that resolve to the same hash values are maintained in a linked list.
- Whenever a collision occurs, the key value is added at the end of the corresponding list.
- Thus, each position in the hash table acts as a header node for a linked list.
- To perform a search operation, the list indicated by the hash function is searched sequentially.

A Separate Chaining Hash Table

- Insert 0, 1, 81, 4, 64, 25, 16, 36, 9, 49.



Insertion

- To perform an insert, we traverse down the appropriate list to check whether the element is already in place (if duplicates are expected, an extra field is usually kept, and this field would be incremented in the event of a match).
- If the element turns out to be new, it is inserted either at the front of the list or at the end of the list, whichever is easiest.

H(Key) = Key % TableSize

Insertion

- Insert 0 H(0) = 0 % 10 = 0
- Insert 1 H(1) = 1 % 10 = 1
- Insert 81 H(81) = 81 % 10 = 1
- The element 81 collides to the same hash value 1. To place the value at this position, perform the following:
  - Traverse the list to check whether it is already present.
  - Since it is not already present, insert at the end of the list.

Insertion

- Similarly, the rest of the elements are inserted.
- Insert 4 H(4) = 4 % 10 = 4
- Insert 64 H(64) = 64 % 10 = 4
- Insert 25 H(25) = 25 % 10 = 5
- Insert 16 H(16) = 16 % 10 = 6
- Insert 36 H(36) = 36 % 10 = 6
- Insert 9 H(9) = 9 % 10 = 9
- Insert 49 H(49) = 49 % 10 = 9

Find

- To perform a find, we use the hash function to determine which list to traverse.
- We then traverse this list in the normal manner, returning the position where the item is found.

Advantage

- More number of elements can be inserted as it uses array of linked lists.
- Collision resolution is simple and efficient.

Disadvantage

- It requires pointers, which occupies more memory space.
- It takes more effort to perform a search, since it takes time to evaluate the hash function and also to traverse the list.

Queries?

Thank You!