Fundamentals of
Data Structures using C

Linear Probing

B.Bhuvaneswaran, AP (SG) / CSE

☎ 9791519152

✉ bhuvaneswaran@rajalakshmi.edu.in

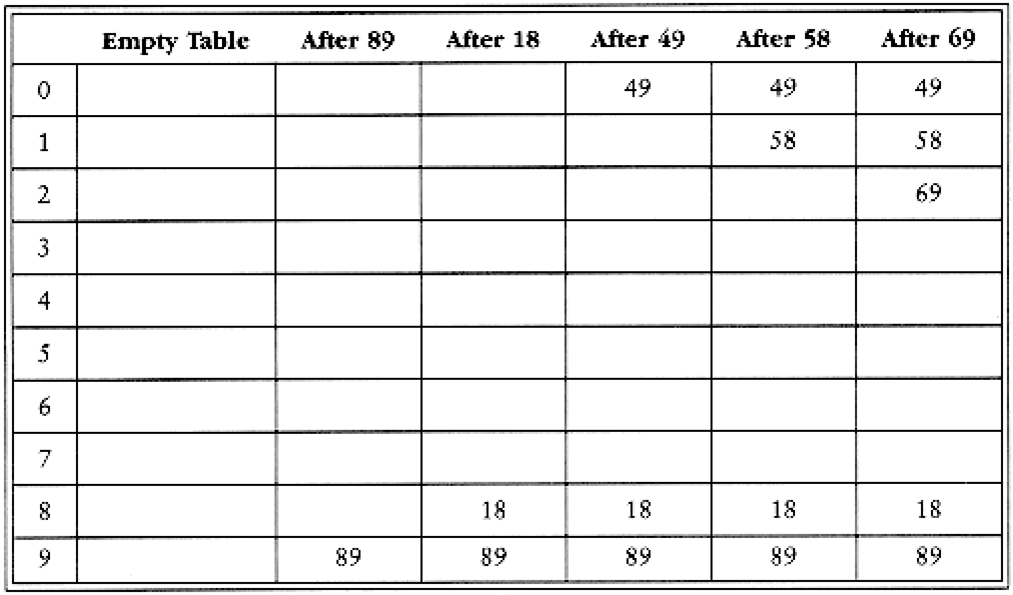
RAJALAKSHMI
ENGINEERING COLLEGE

Introduction

- In linear probing, F is a linear function of i, typically F(i) = i.
- This amounts to trying cells sequentially (with wraparound) in search of an empty cell.
- $h_i(X) = (Hash(X) + i) \% N$
- $h_0(X) = Hash(X) \% N$
- $h_1(X) = (Hash(X) + 1) \% N$
- $h_2(X) = (Hash(X) + 2) \% N$

Insertion

- Figure shows the result of inserting keys {89, 18, 49, 58, 69} into a hash table using the same hash function as before and the collision resolution strategy, F(i) = i.



	Empty Table	After 89	After 18	After 49	After 58	After 69
0				49	49	49
1					58	58
2						69
3						
4						
5						
6						
7						
8			18	18	18	18
9		89	89	89	89	89

Insertion

- The first collision occurs when 49 is inserted; it is put in the next available spot, namely spot 0, which is open.
- 58 collides with 18, 89, and then 49 before an empty cell is found three away.
- The collision for 69 is handled in a similar manner.

Insertion

- As long as the table is big enough, a free cell can always be found, but the time to do so can get quite large.
- Worse, even if the table is relatively empty, blocks of occupied cells start forming.
- This effect, known as primary clustering, means that any key that hashes into the cluster will require several attempts to resolve the collision, and then it will add to the cluster.

Advantage

- It doesn't require pointers.

Disadvantage

- It forms clusters, which degrades the performance of the hash table for storing and retrieving.

Queries?

Thank You!