LINEAR PROBING

• Probing is the process of getting next available hash table array cell.

NGINEER

• In linear probing, F(i) is a linear function of i. That means alternative locations are searched in a sequential manner. Thus,

$$F(i)=i$$

Example:

Insert the keys {89, 18, 49, 58, 69} into the hash table.

Empty Table After 89 After 18 After 49 After 58 After 69 0 49 49 49 49 1 58 58 58 2 69 69 69 5 6 7 7 8 18 18 18 9 89 89 89 89						// <u> </u>	Г
Table 0 49 49 49 1 58 58 2 69 3 4 5 69 5 6 7 8 18 18 18 18		Empty	After 80	After 18	After 10	After 58	After 60
1 58 58 2 69 3 69 5 6 7 8 18 18 18 18		Table	Aitci 69	Alter 18	AILLI 49	Alter 58	Aitel 09
2 3 4 5 6 7 8 18 18 18 18	0	- 4	."/ / {	**	49	49	
3 4 5 6 7 8 18 <t< td=""><td>1</td><td>7 :</td><td>{</td><td>FO.</td><td>1</td><td>58</td><td>58</td></t<>	1	7 :	{	FO.	1	58	58
4 5 6 7 8 18 18 18 18	2	6	5 /	III O	00	41	69
5 6 7 8 18 18 18	3	E	3\ /				9
6 7 8 18 18 18 18	4	1	Ē \\ [(開)	0/
7 8 18 18 18	5		8:1/1				9
8 18 18 18	6		*			*	
	7					INAFI	
9 89 89 89 89	8			18	18	18	18
	9		89	89	89	89	89

 $h_i(X) = (Hash(X) + F(i)) mod Table size$

Hash $(89) = 89 \mod 10 = 9$ (No collision)

Hash $(18) = 18 \mod 10 = 8$ (No collision)

Hash (49) =49 mod
$$10 = 9$$
 (Collision Occurred)
$$h_i (49) = (Hash (49) + F(1)) \text{ mod } 10$$

$$= (9 + 1) \text{ mod } 10 = 10 \text{ mod } 10 = 0 \text{(No collision)}$$
Hash (58) =58 mod $10 = 8$ (Collision occurred)

$$= (8 + 1) \mod 10 = 9 \mod 10 = 9 (Collision occurred)$$

$$h_i (58) = (Hash (58) + F(2)) \mod 10$$

$$= (8 + 2) \mod 10 = 10 \mod 10 = 0 (Collision occurred)$$

$$h_i (58) = (Hash (58) + F(3)) \mod 10$$

$$= (8 + 3) \mod 10 = 11 \mod 10 = 1 (No Collision)$$

Hash
$$(69) = 69 \mod 10 = 9$$
 (Collision Occurred)

 $h_i(58) = (Hash(58) + F(1)) \mod 10$

$$h_i$$
 (69) = (Hash (69) + F(1)) mod 10
= (9+1) mod 10 = 10 mod 10 = 0(Collision occurred)

$$h_i$$
 (69) = (Hash (69) + F(2)) mod 10
= (9 + 2) mod 10 = 11 mod 10 = 1(Collision occurred)

$$h_i(69) = (Hash (69) + F(3)) \mod 10$$

= (9 + 3) mod 10 = 12 mod 10 = 2(No Collision)

OBSERVE OPTIMIZE OUTSPREAD