

# ALG 13b

## Coalesced Hashing

### Examples

**LISCH (late insert standard coalesced hashing)**

**EISCH (early insert standard coalesced hashing)**

**LICH (late insert coalesced hashing)**

**EICH (early insert coalesced hashing)**

**VICH (variable insert coalesced hashing)**

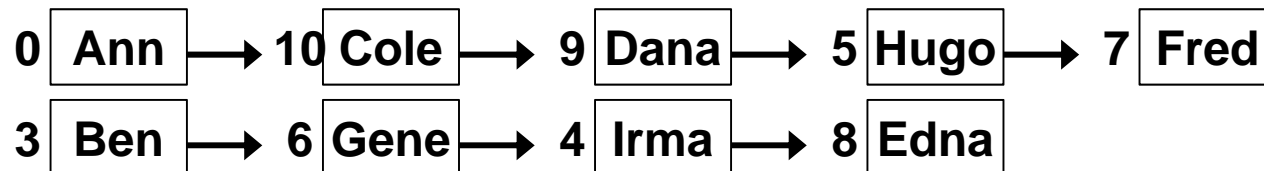
## Coalesced hashing

It is a collision resolving method, it is not related to any particular hash function  $h(k)$ .

A synonym (after a collision) is stored in a one-way linked list of synonyms. More mutually disjoint lists exist in the table. Each key belongs to some list. Each table entry contains a reference to the next entry in the list.

Searching follows the same rules as inserting. Basically, it is always just a sequential linked list search.

0	Ann	10	
1			
2	Ben	6	
3			
4	Irma	8	
5	Hugo	7	
6	Gene	4	
7	Fred	--	
8	Edna	--	
9	Dana	5	
10	Cole	9	



## LISCH (late insert standard coalesced hashing)

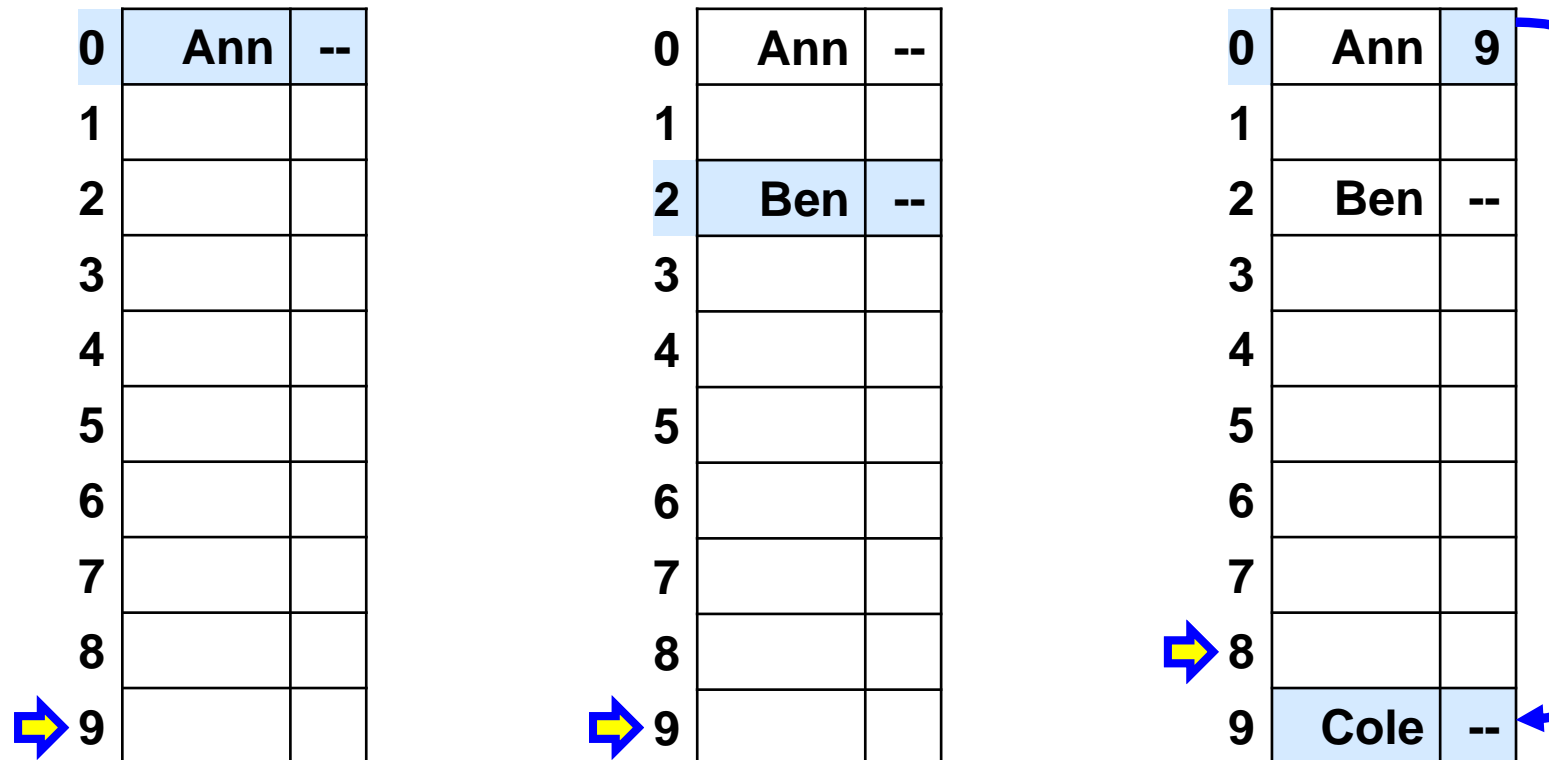
Hash function  $h$ , key  $k$ .  
 Position  $p := h(d)$ ;  
 Search the list starting from position  $p$  and if  $k$  is not found store  $k$  at first free position from the table bottom. Also, append  $k$  to the end of the searched list of synonyms.

Pointer to the first free slot from the table end. It is updated (moved up) after each insert.

	Name	Next
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		

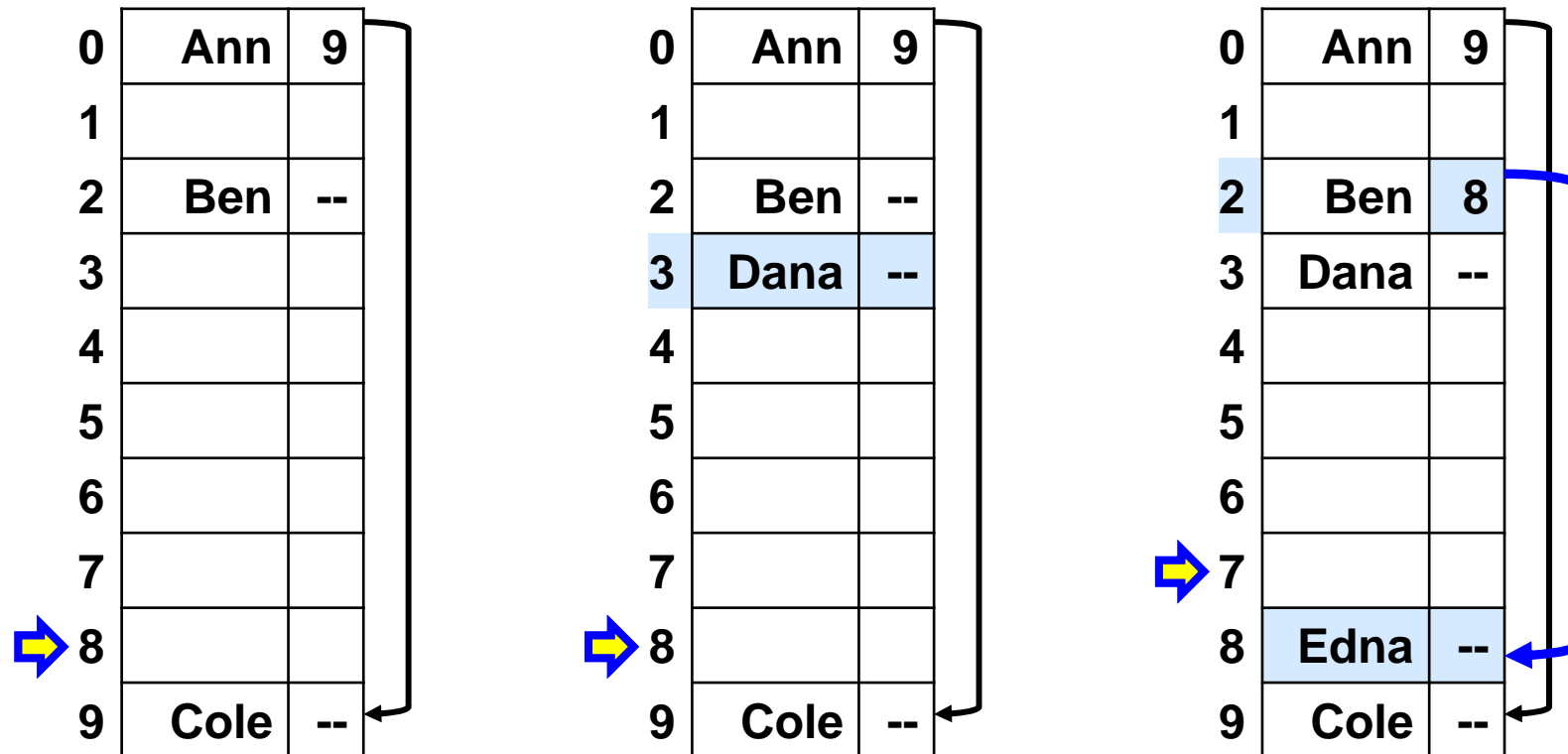
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
$h(\text{data})$	0	2	0	3	2	9	0	8	7

# LISCH (late insert standard coalesced hashing)



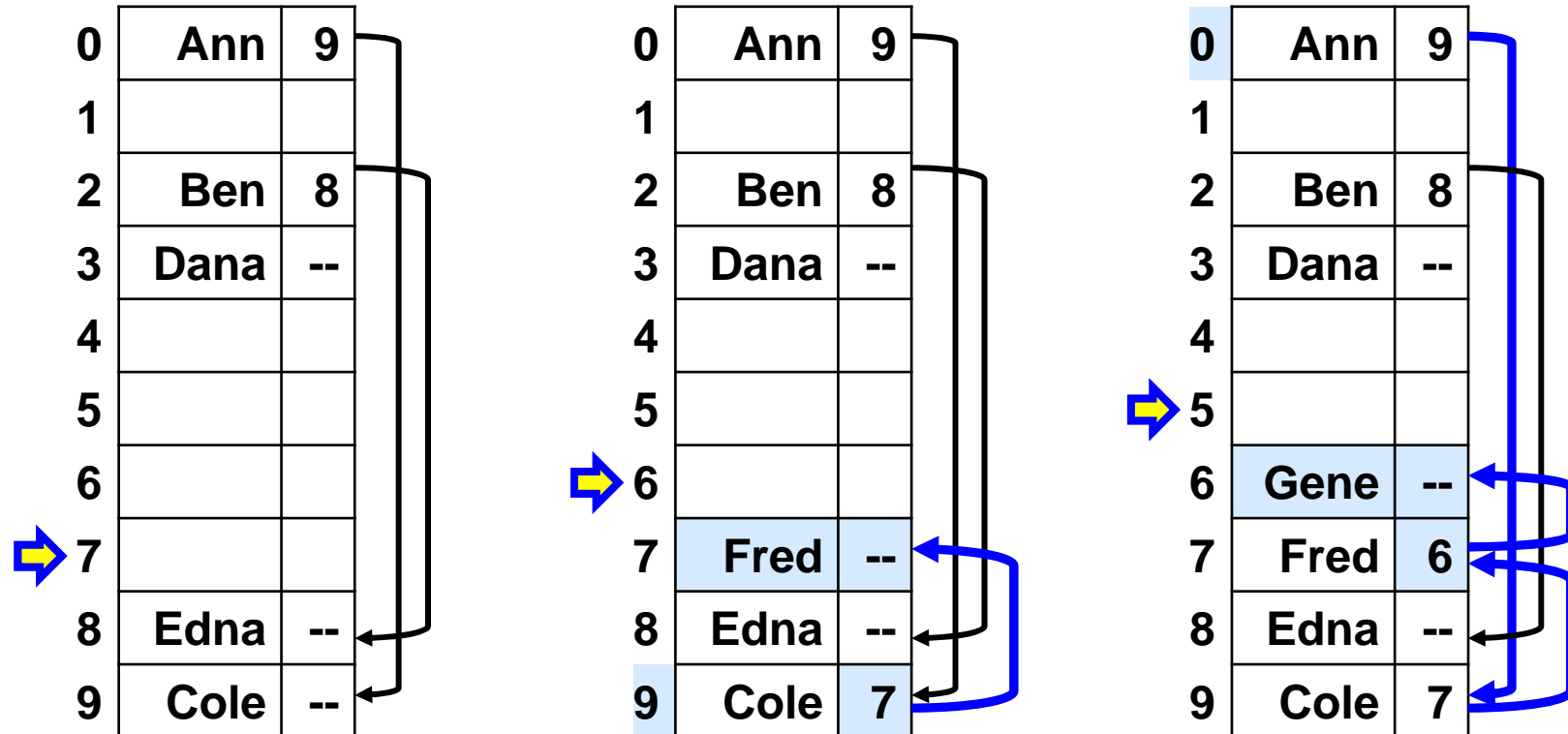
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	3	2	9	0	8	7

# LISCH (late insert standard coalesced hashing)



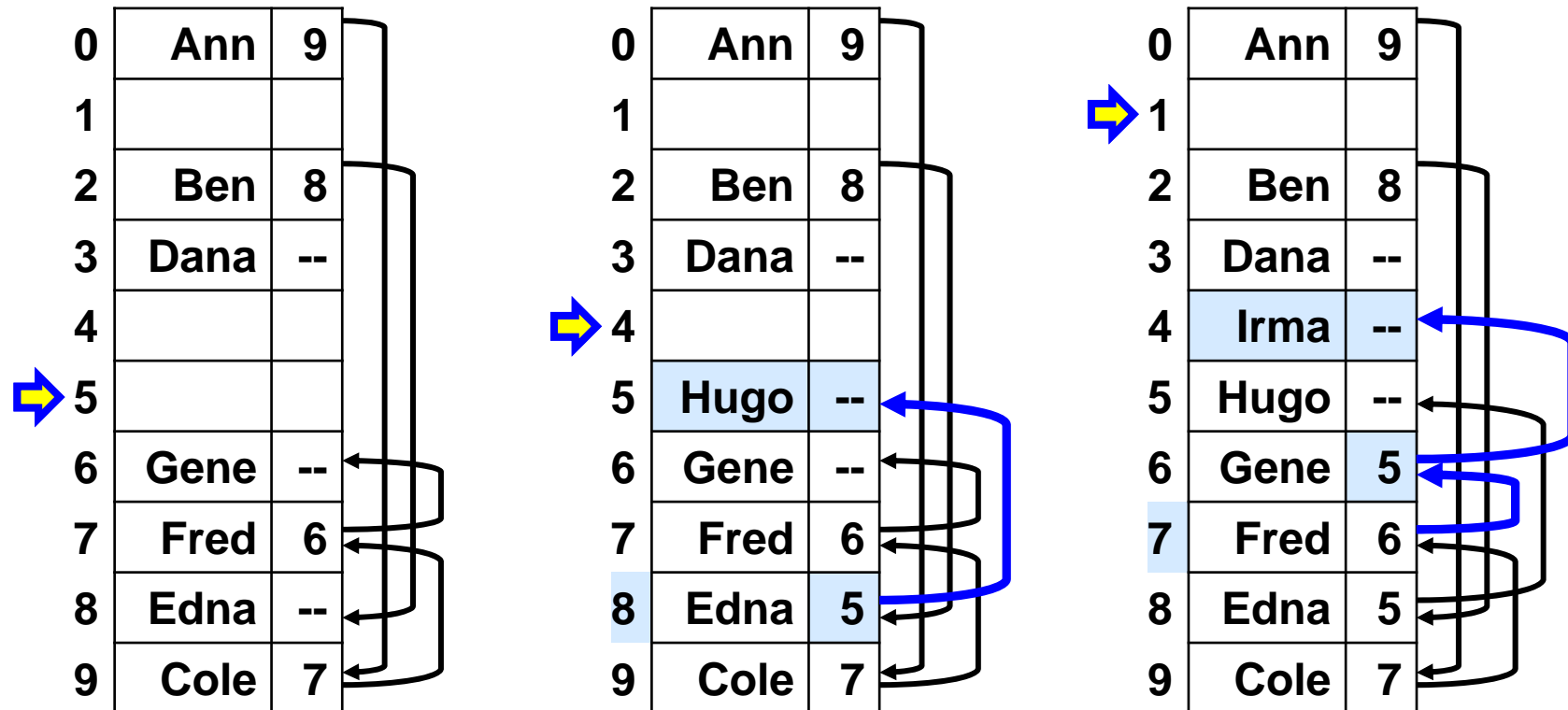
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	3	2	9	0	8	7

# LISCH (late insert standard coalesced hashing)



data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	3	2	9	0	8	7

# LISCH (late insert standard coalesced hashing)



data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	3	2	9	0	8	7

## EISCH (early insert standard coalesced hashing)

Hash function  $h$ , key  $k$ .  
 Position  $p := h(d)$ ;  
 Search the list starting from position  $p$  and if  $k$  is not found store  $k$  at first free position from the table bottom. Also, insert  $k$  into searched list just behind the first colliding synonym.

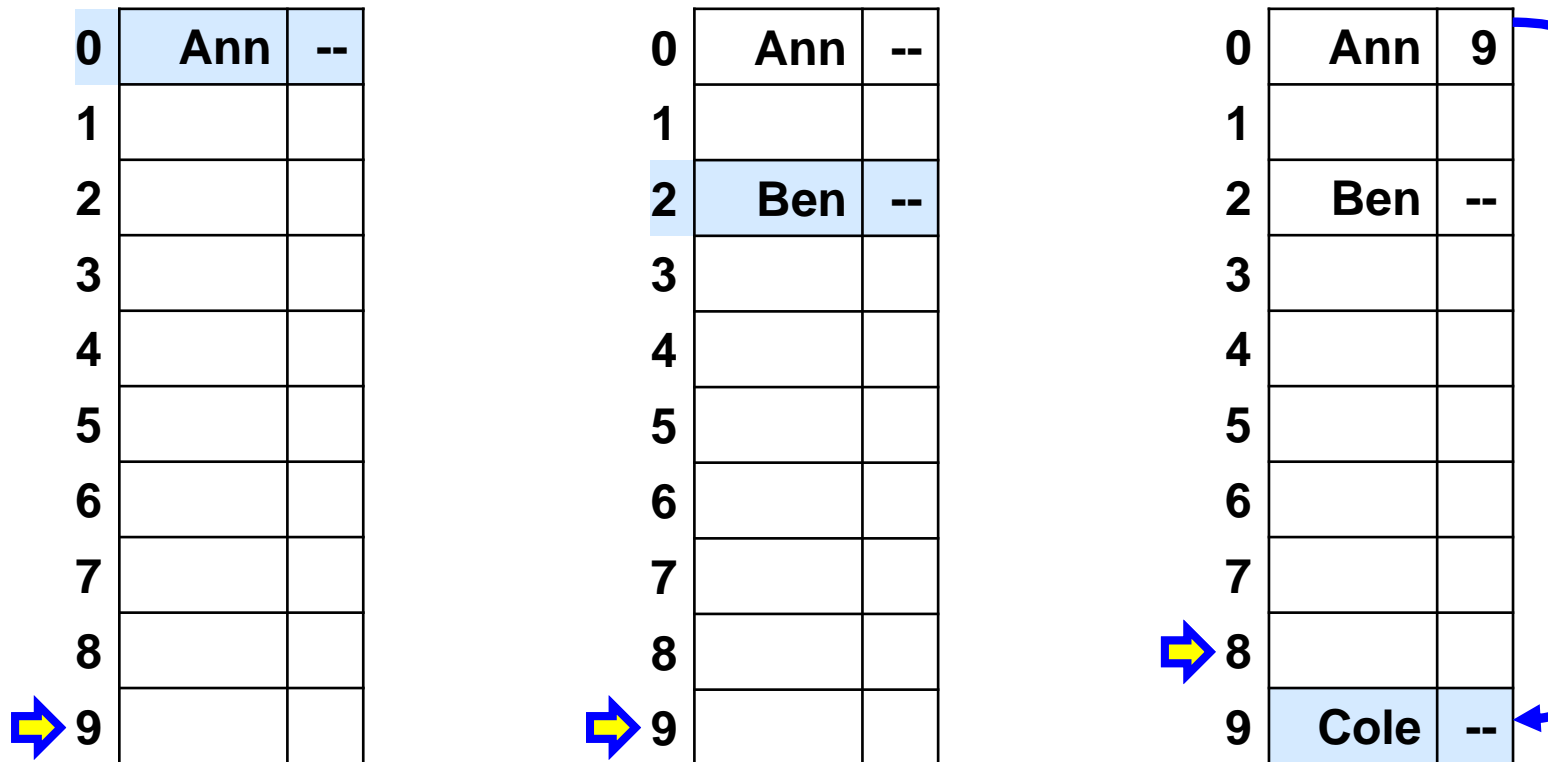
Pointer to the first free slot from the table end. It is updated (moved up) after each insert.

	Name	Next
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		

data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
$h(\text{data})$	0	2	0	3	2	9	0	2	6

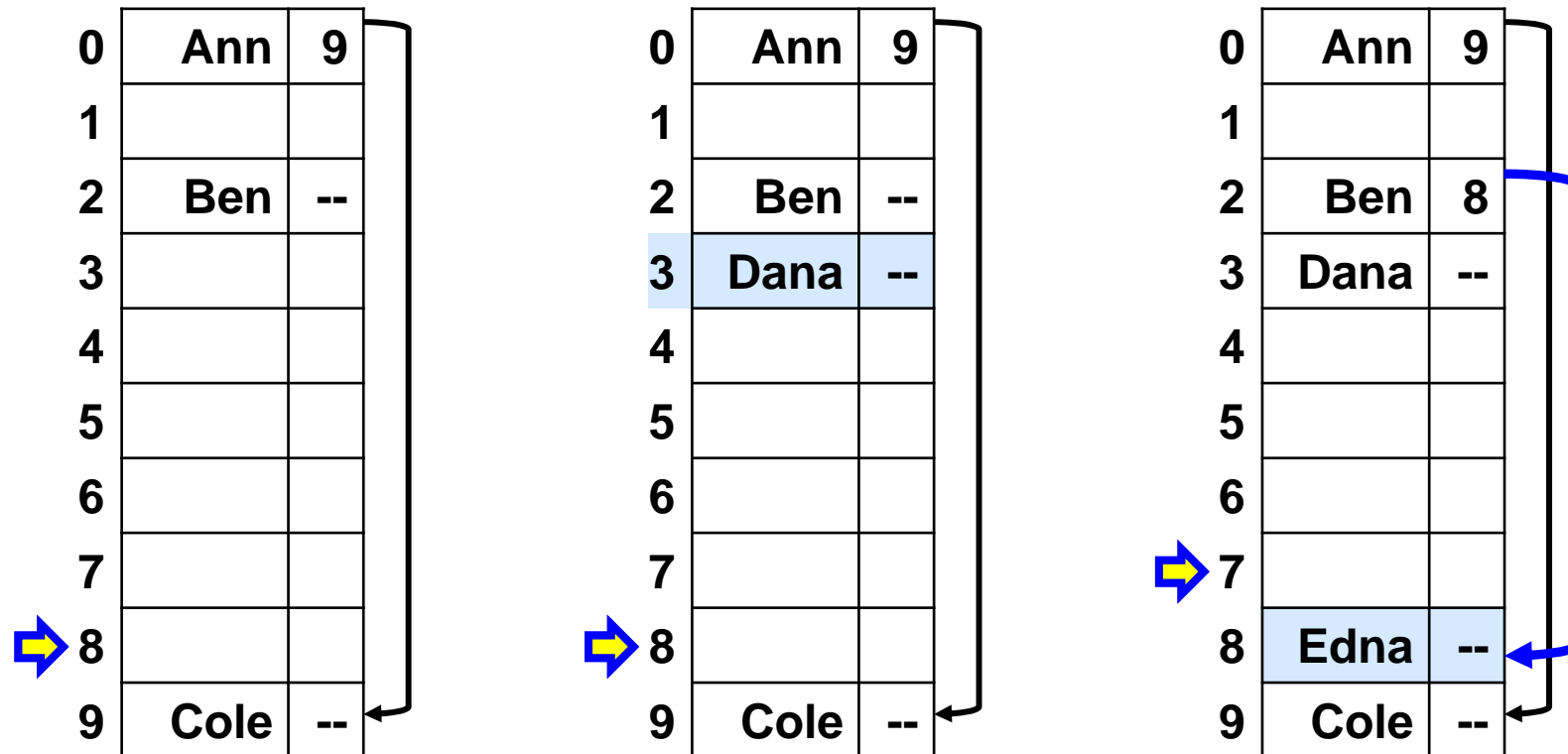


## EISCH (early insert standard coalesced hashing)



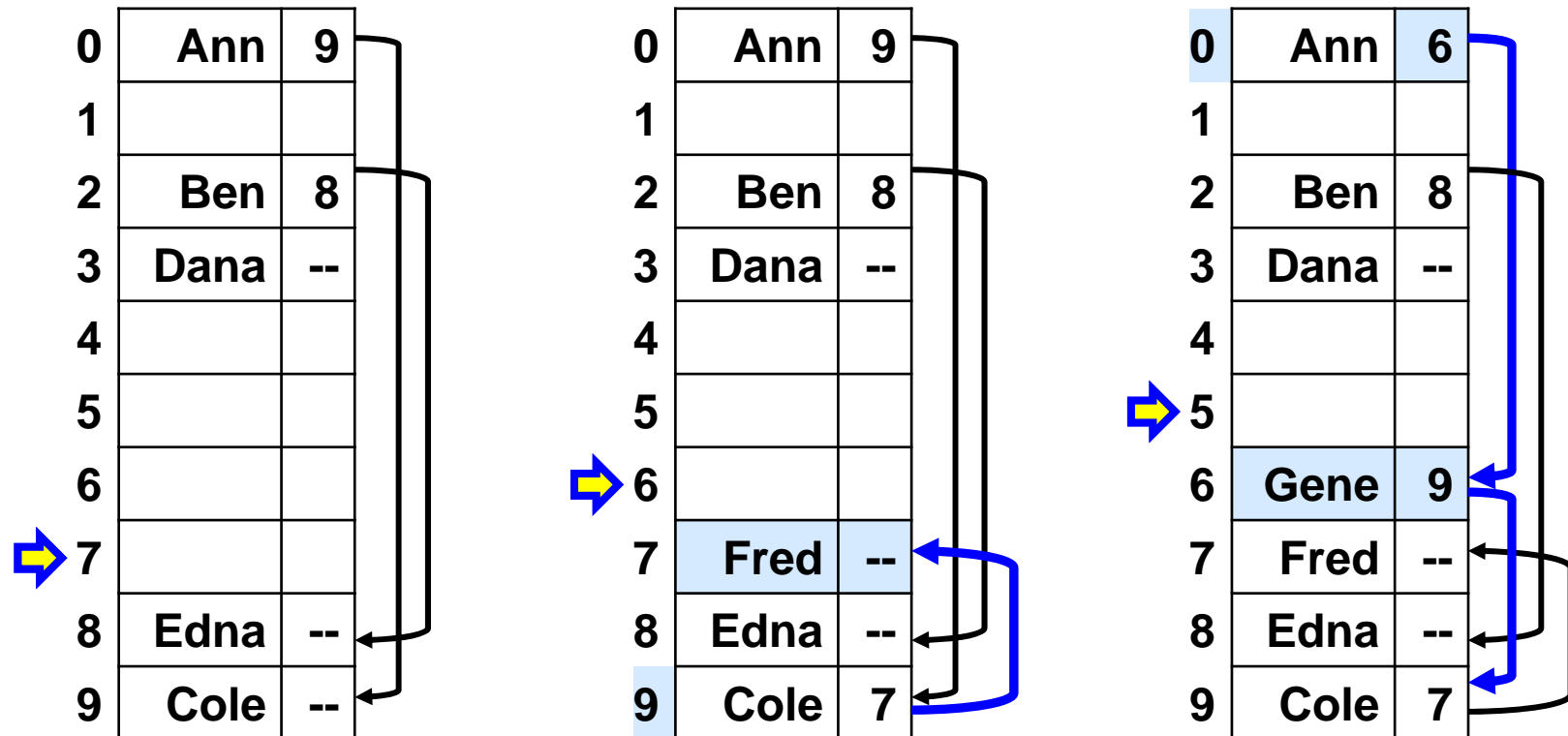
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	3	2	9	0	2	6

## EISCH (early insert standard coalesced hashing)



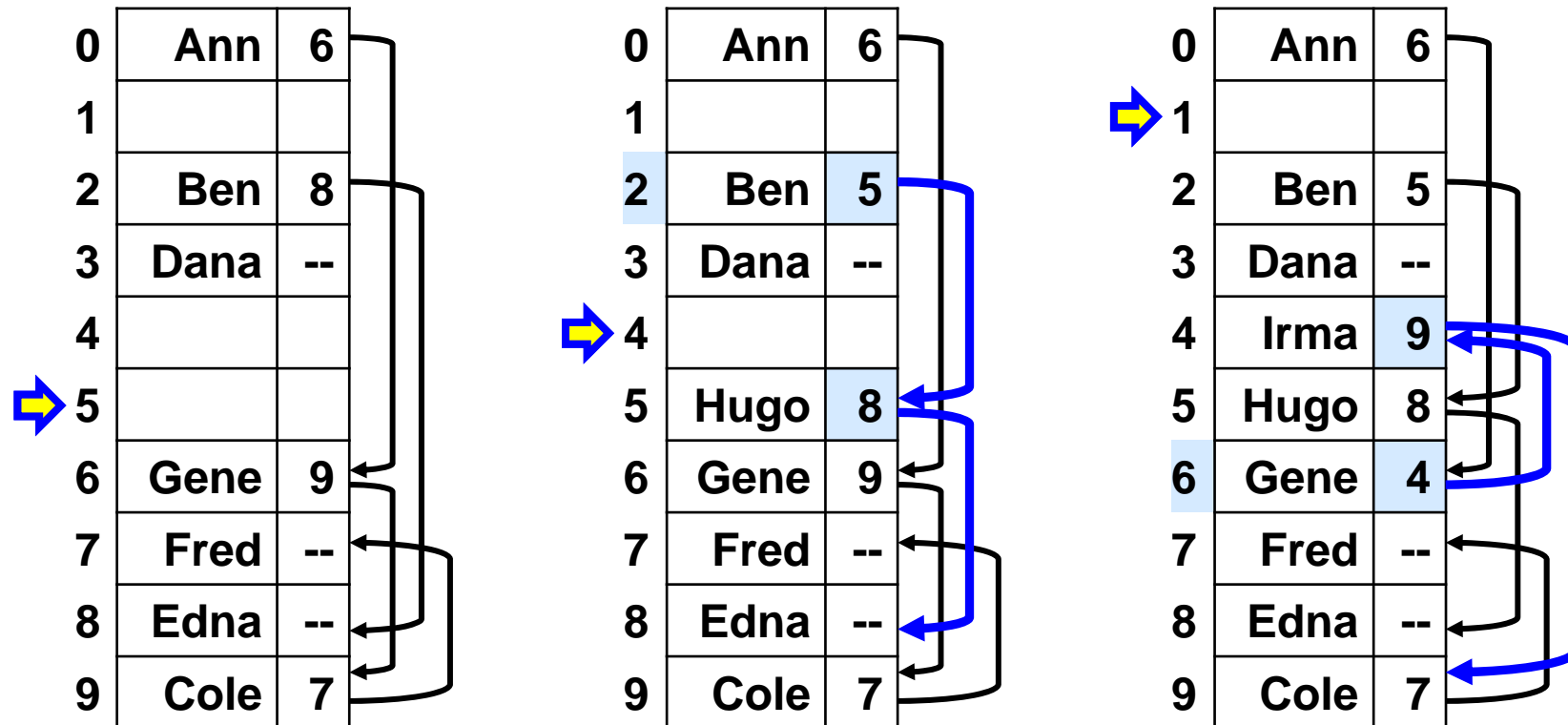
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	3	2	9	0	2	6

## EISCH (early insert standard coalesced hashing)



data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	3	2	9	0	2	6

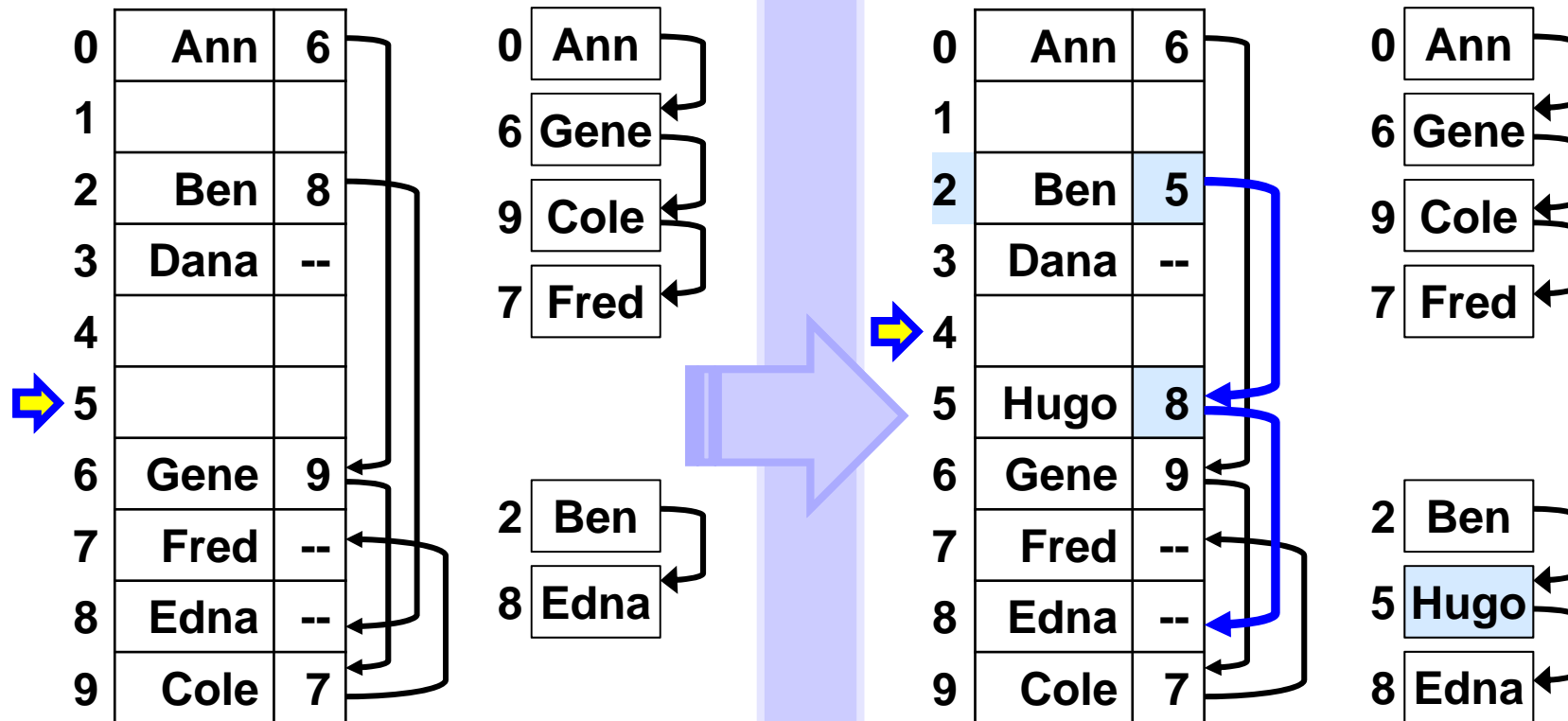
## EISCH (early insert standard coalesced hashing)



data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
$h(\text{data})$	0	2	0	3	2	9	0	2	6

# EISCH (early insert standard coalesced hashing)

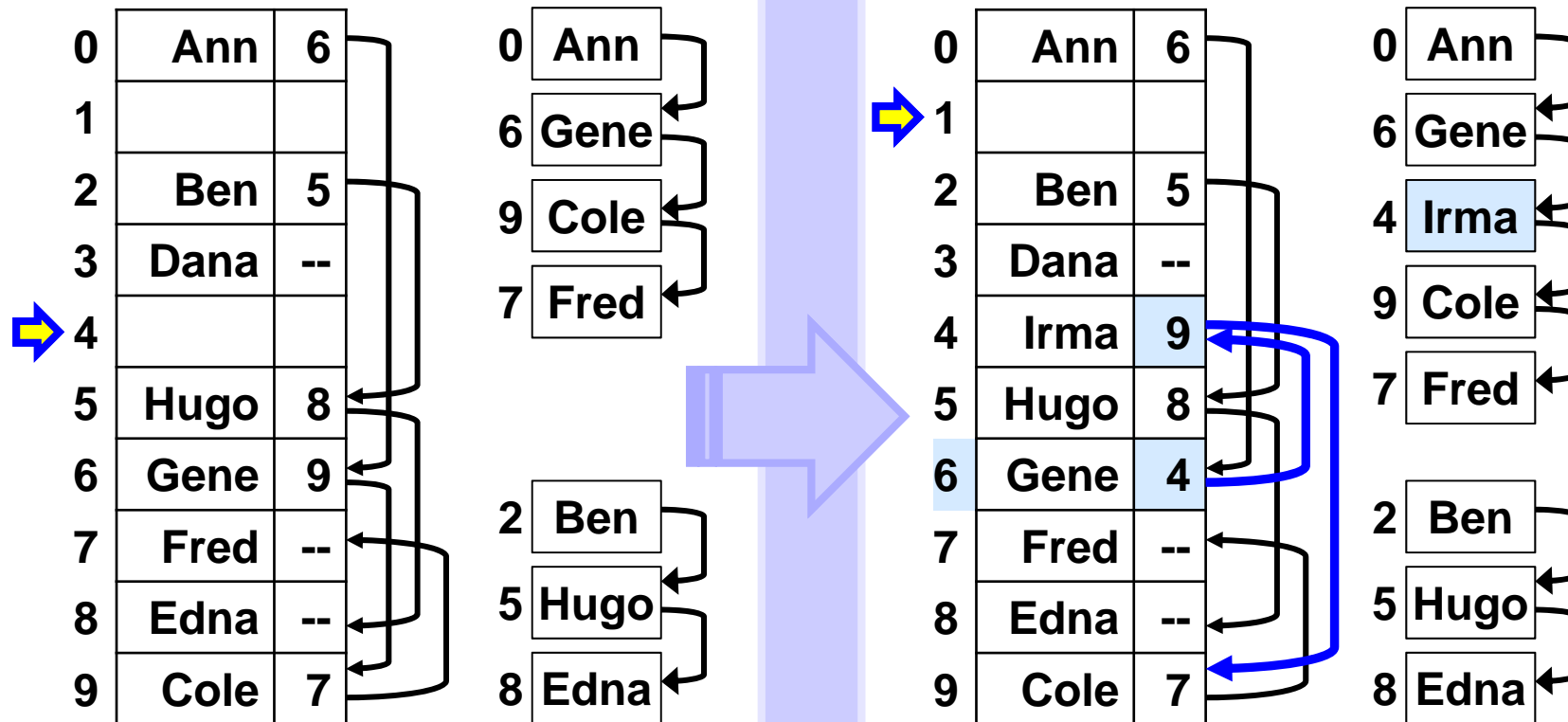
Insert(Hugo)



data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	3	2	9	0	2	6

# EISCH (early insert standard coalesced hashing)

Insert(Irma)



data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
$h(\text{data})$	0	2	0	3	2	9	0	2	6

## Coalesced hashing with added cellar space

To reduce the coalescing the table is expanded by non-addressable space - cellar.

The cellar is at the end of the table and has the same structure as the table.

Algorithms LICH a EICH are analogous variants of algorithms LISCH a EISCH expanded by the cellar.

Algorithm VICH (variable insert coalesced hashing) adds the colliding key to the end of collision chain if the chain ends in the cellar. If the chain ends in outside the cellar the key is inserted into the chain at the point where the chain leaves the cellar.

## LICH (late insert coalesced hashing)

0	Ann	--
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

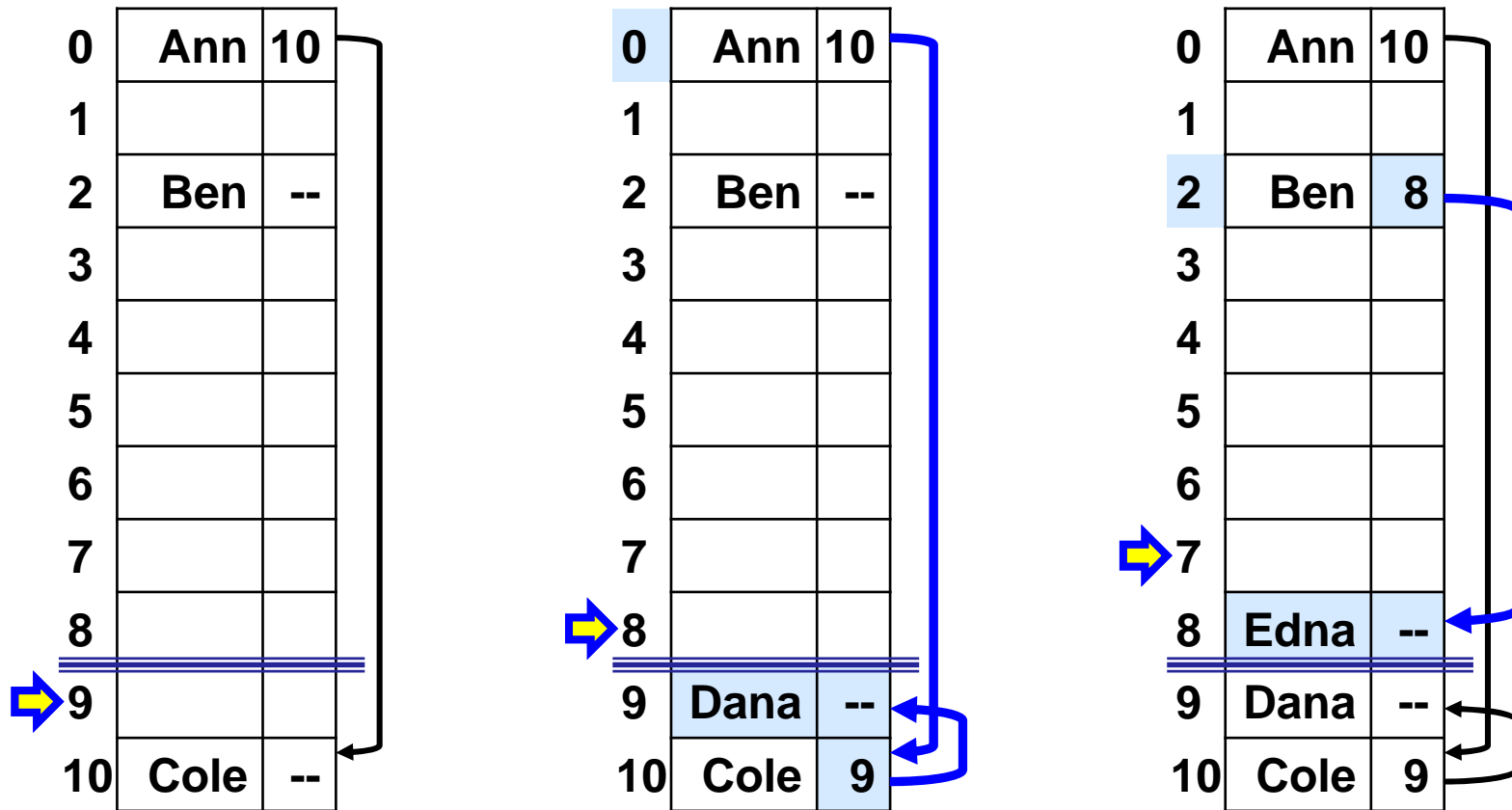
0	Ann	--
1		
2	Ben	--
3		
4		
5		
6		
7		
8		
9		
10		

0	Ann	10
1		
2	Ben	--
3		
4		
5		
6		
7		
8		
9		
10	Cole	--

data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	7	5	8

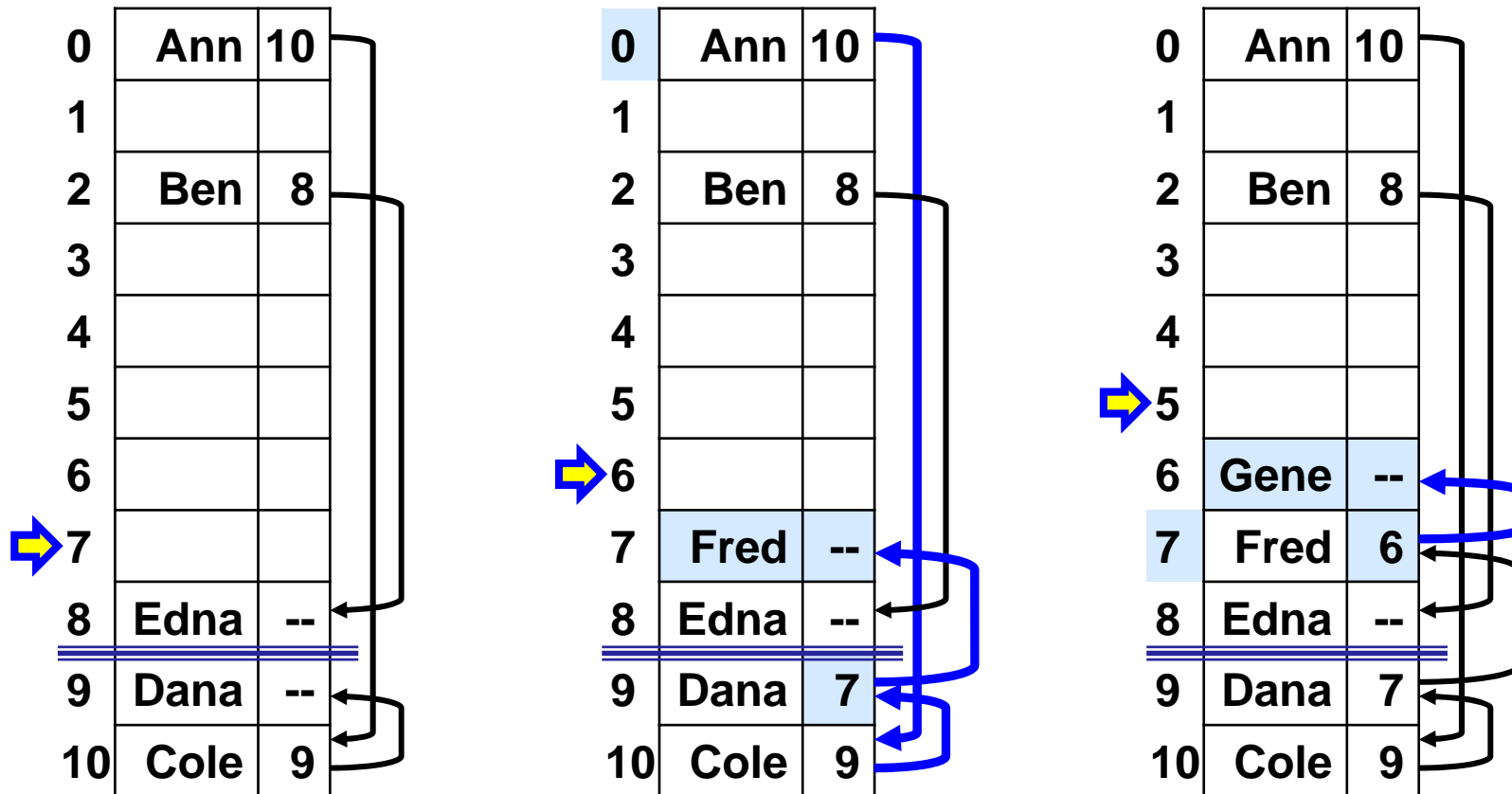


## LICH (late insert coalesced hashing)



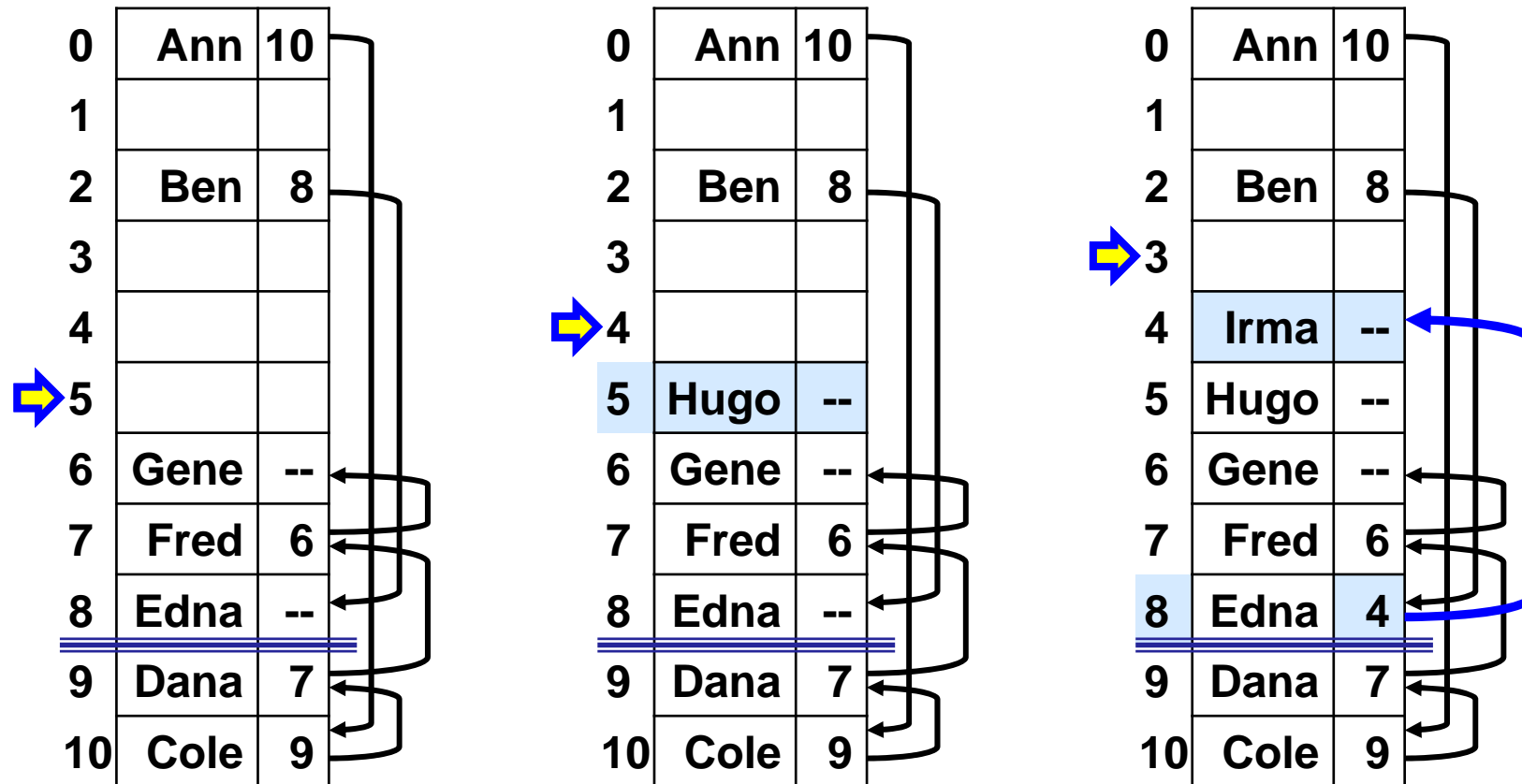
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	7	5	8

## LICH (late insert coalesced hashing)



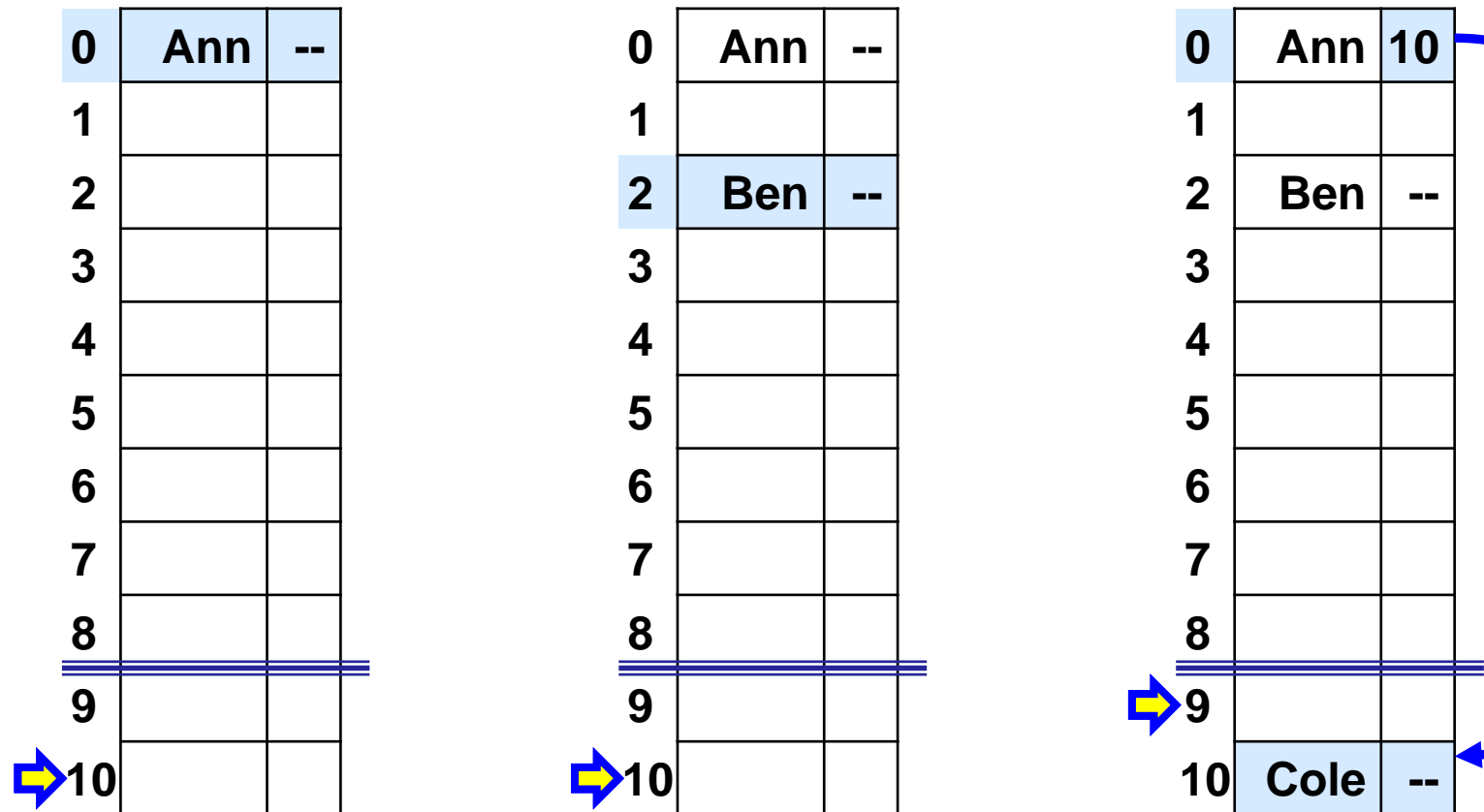
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
$h(\text{data})$	0	2	0	0	2	0	7	5	8

## LICH (late insert coalesced hashing)



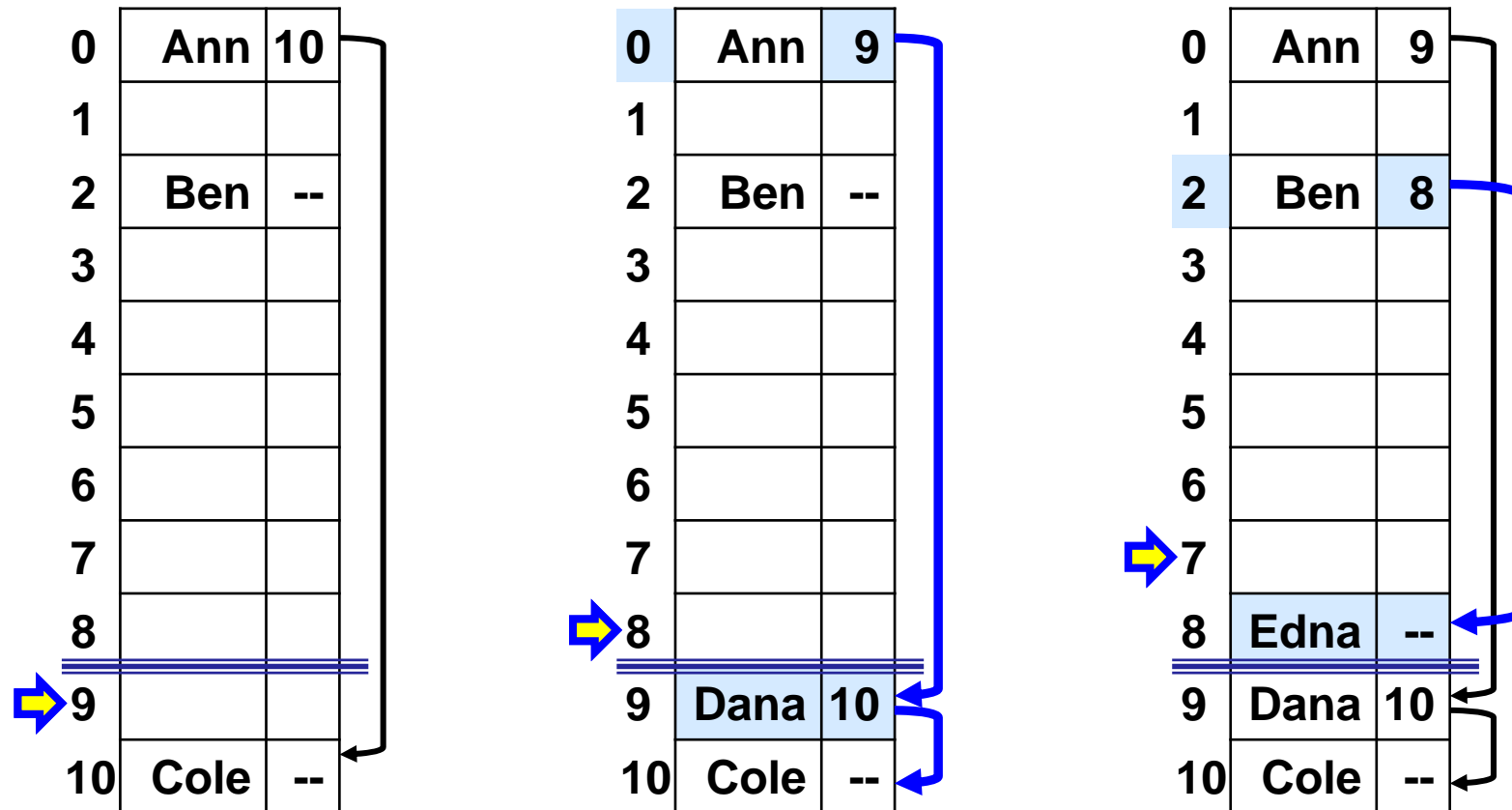
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	7	5	8

## EICH (early insert coalesced hashing)



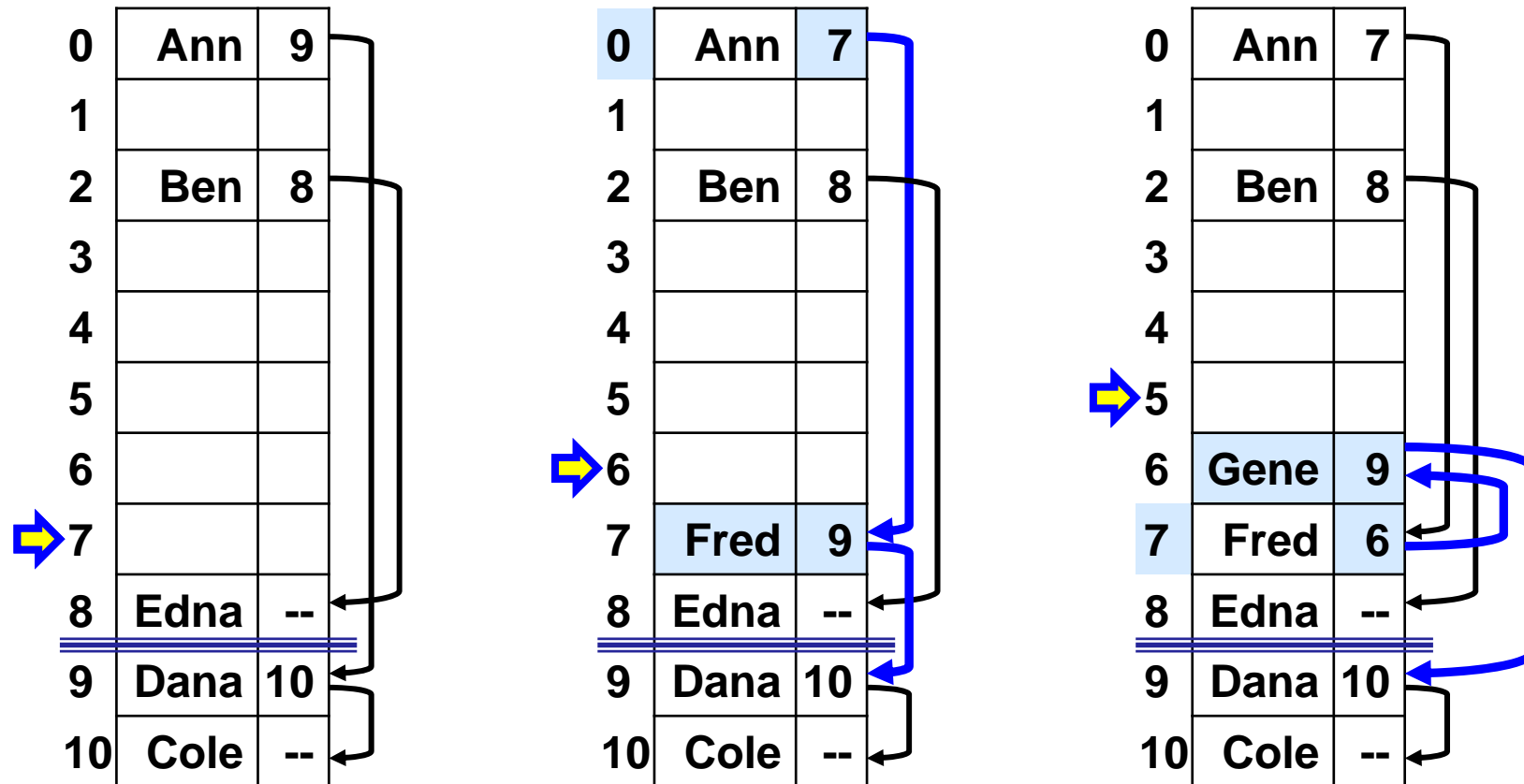
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	7	0	8

## EICH (early insert coalesced hashing)



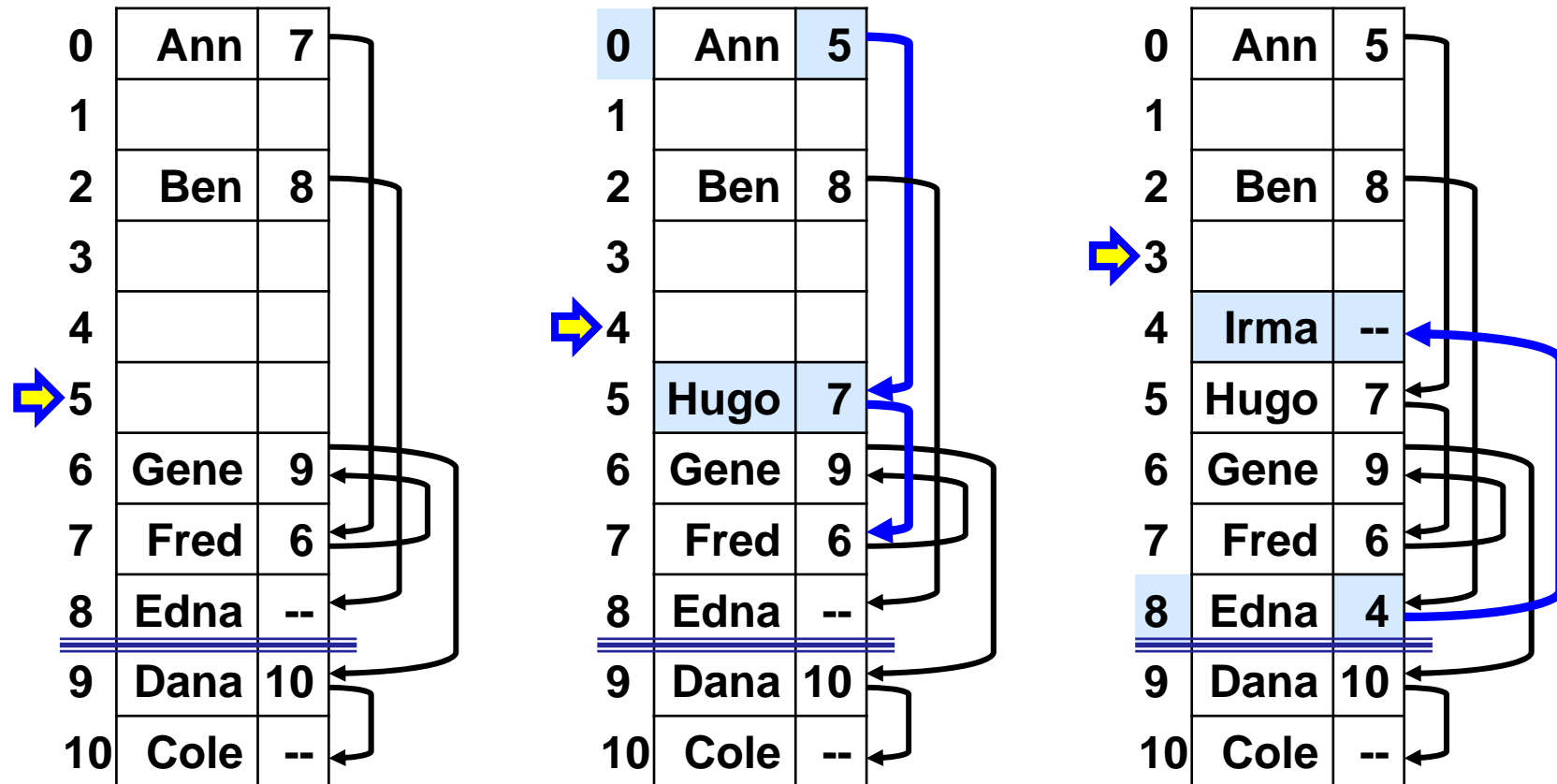
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	7	0	8

## EICH (early insert coalesced hashing)



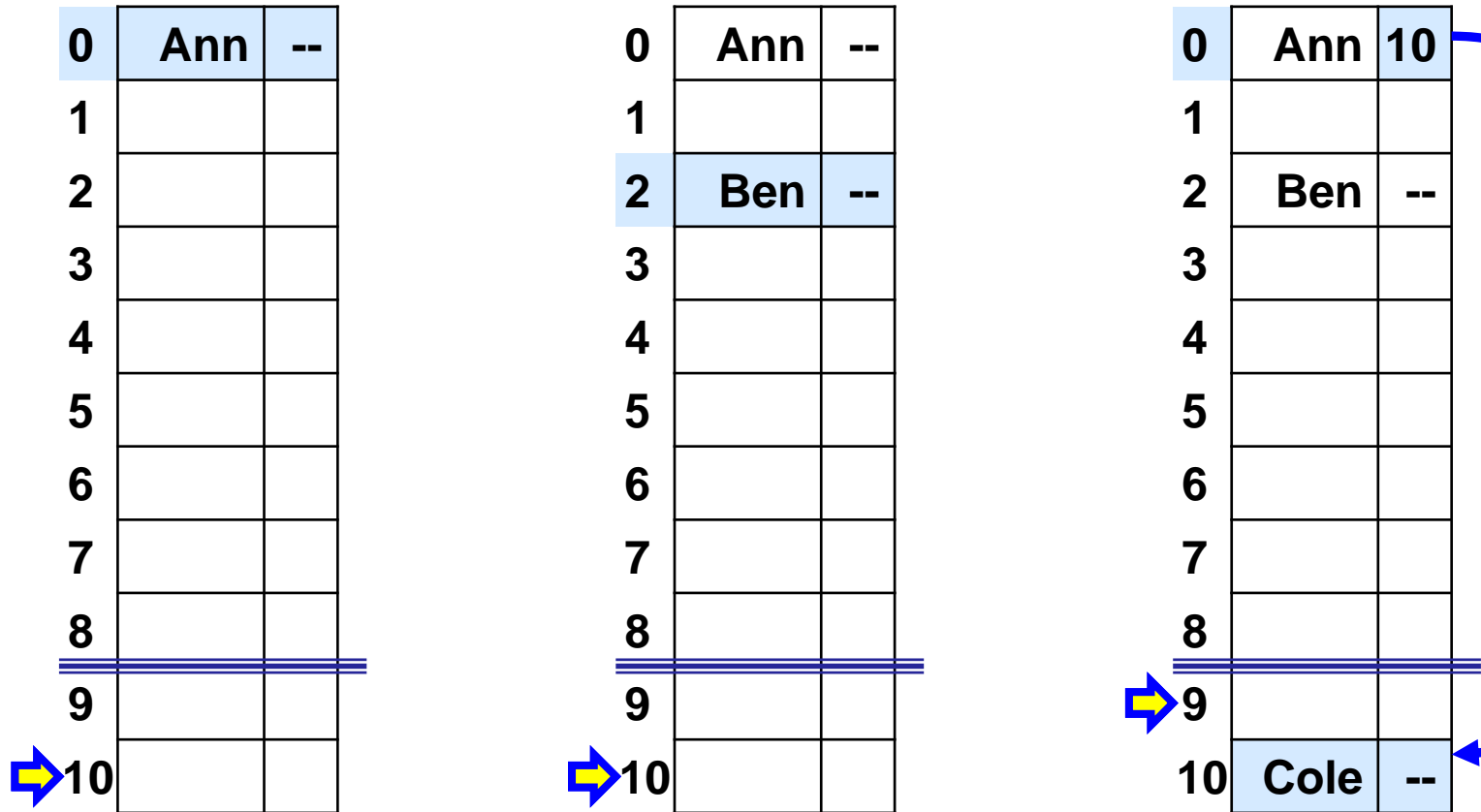
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
$h(\text{data})$	0	2	0	0	2	0	7	0	8

## EICH (early insert coalesced hashing)



data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	7	0	8

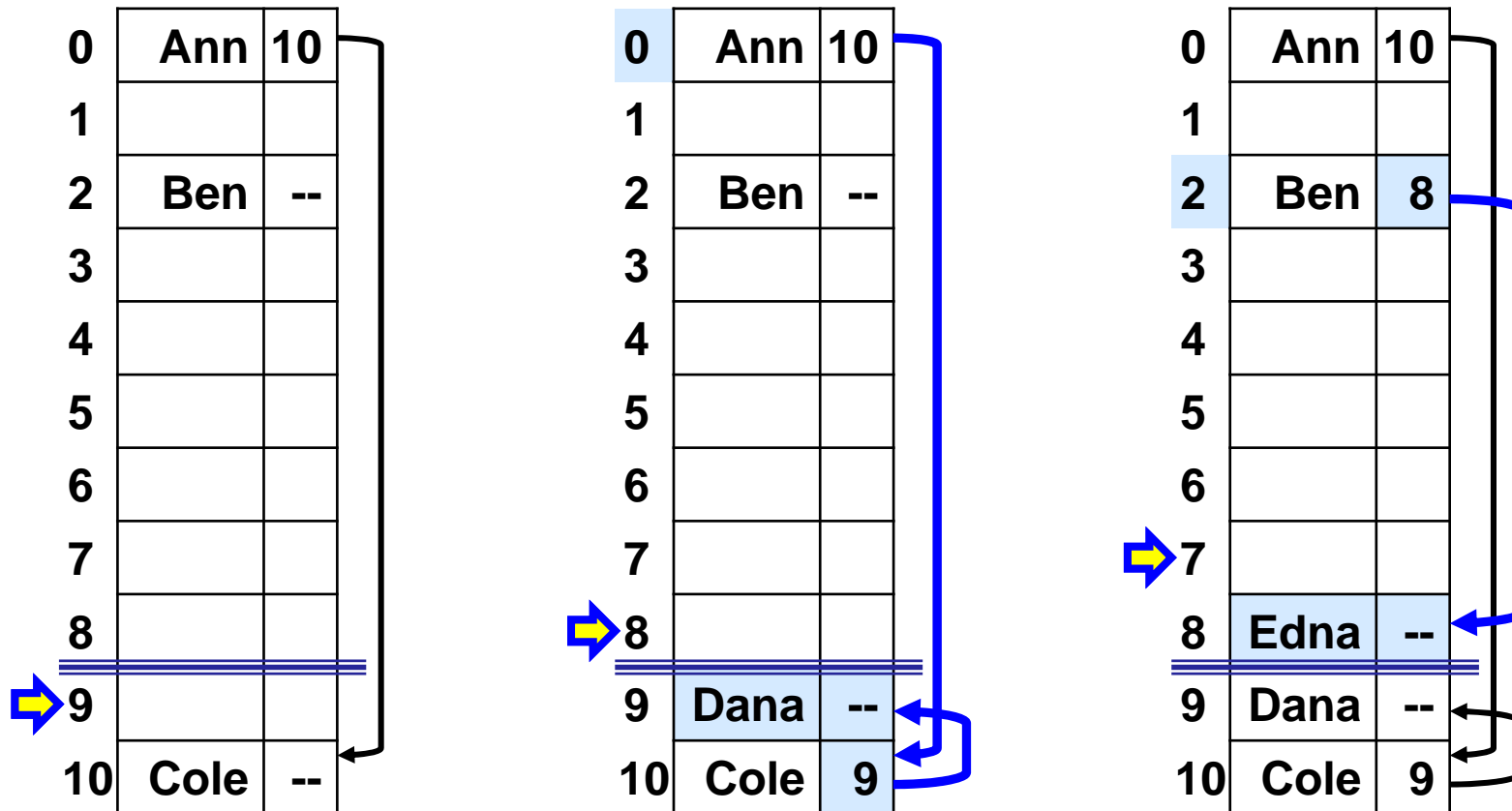
## VICH (variable insert coalesced hashing)



data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	2	0	6

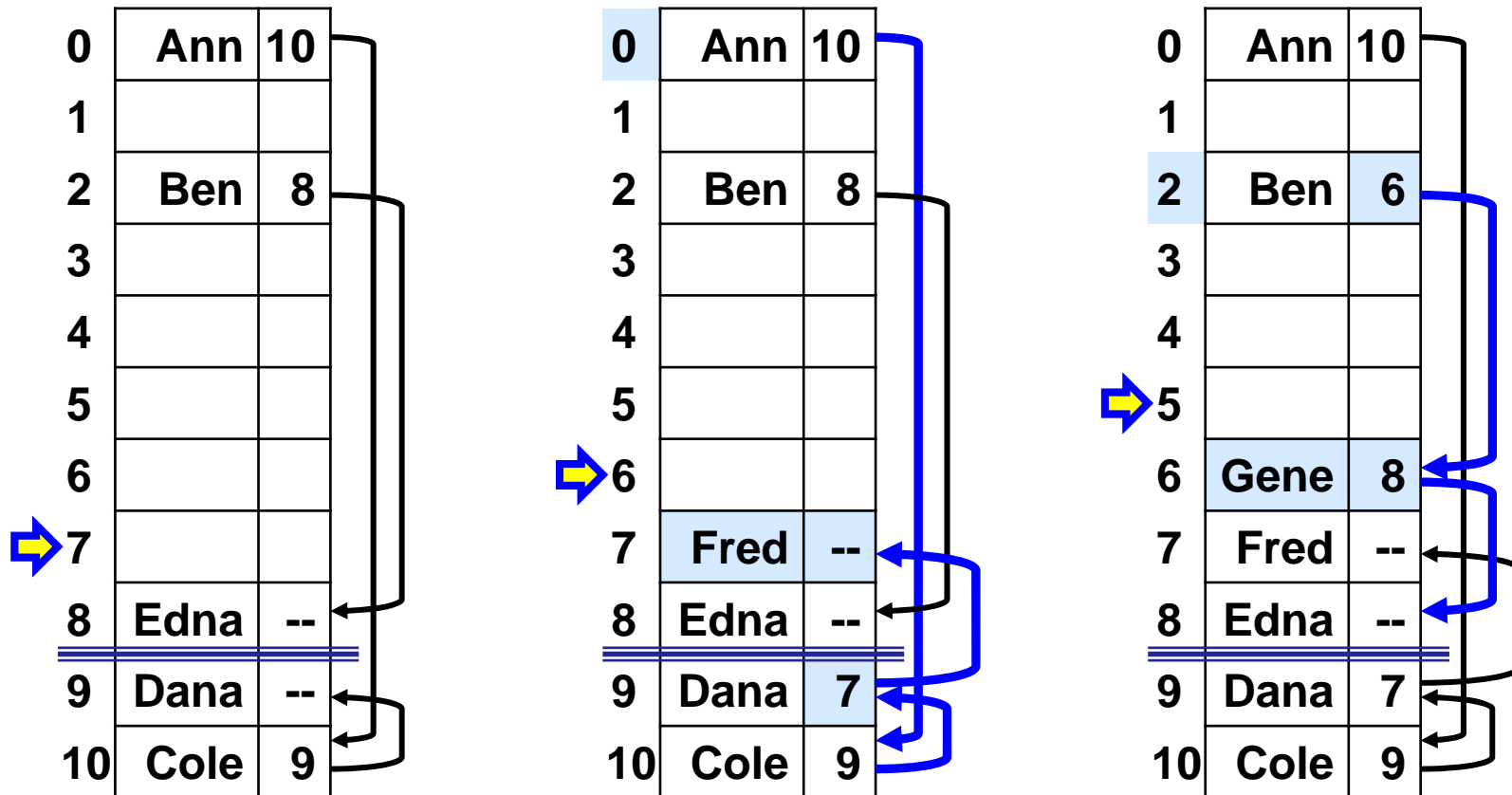


## VICH (variable insert coalesced hashing)



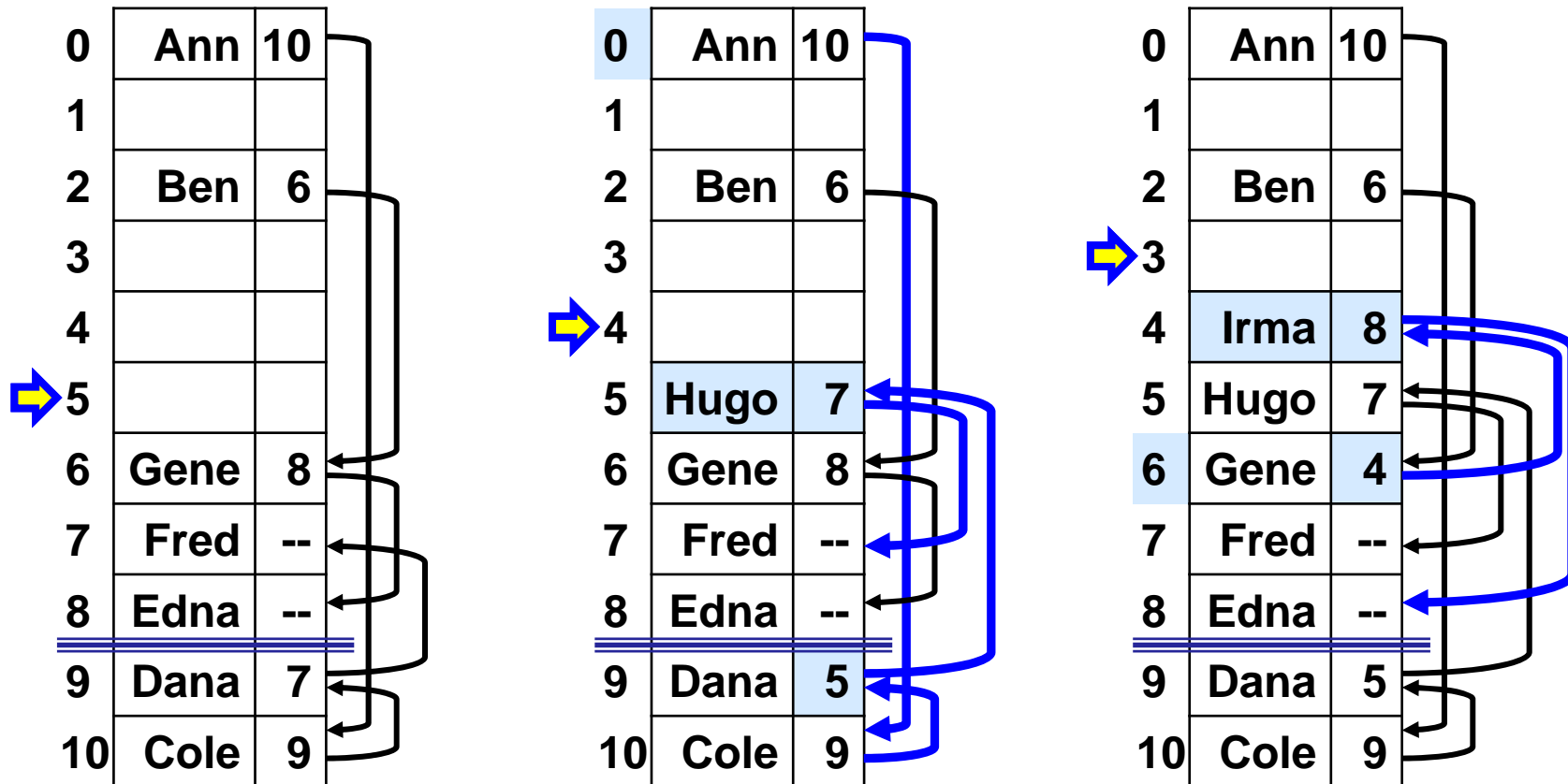
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	2	0	6

## VICH (variable insert coalesced hashing)



data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
$h(\text{data})$	0	2	0	0	2	0	2	0	6

## VICH (variable insert coalesced hashing)



data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
$h(\text{data})$	0	2	0	0	2	0	2	0	6