

# ALG 13b

## Srůstající hashování

### Ukázky

**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)**

## Srůstající hashování -- coalesced hashing

Jde o metodu řešení kolizí, nezáleží na konkrétní podobě hashovací funkce  $h(k)$ .

Synonyma (po kolizi) se ukádají do jednosměrného spojového seznamu synonym. Všechny seznamy jsou "propleteně" uloženy přímo v tabulce.

Tabulka ke každému klíči obsahuje ukazatel na další klíč v seznamu.

Každý klíč je součástí některého seznamu synonym.

Při vyhledávání se postupuje stejně jako při vkládání, v podstatě jde o lineární prohledávání spojového seznamu.

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)

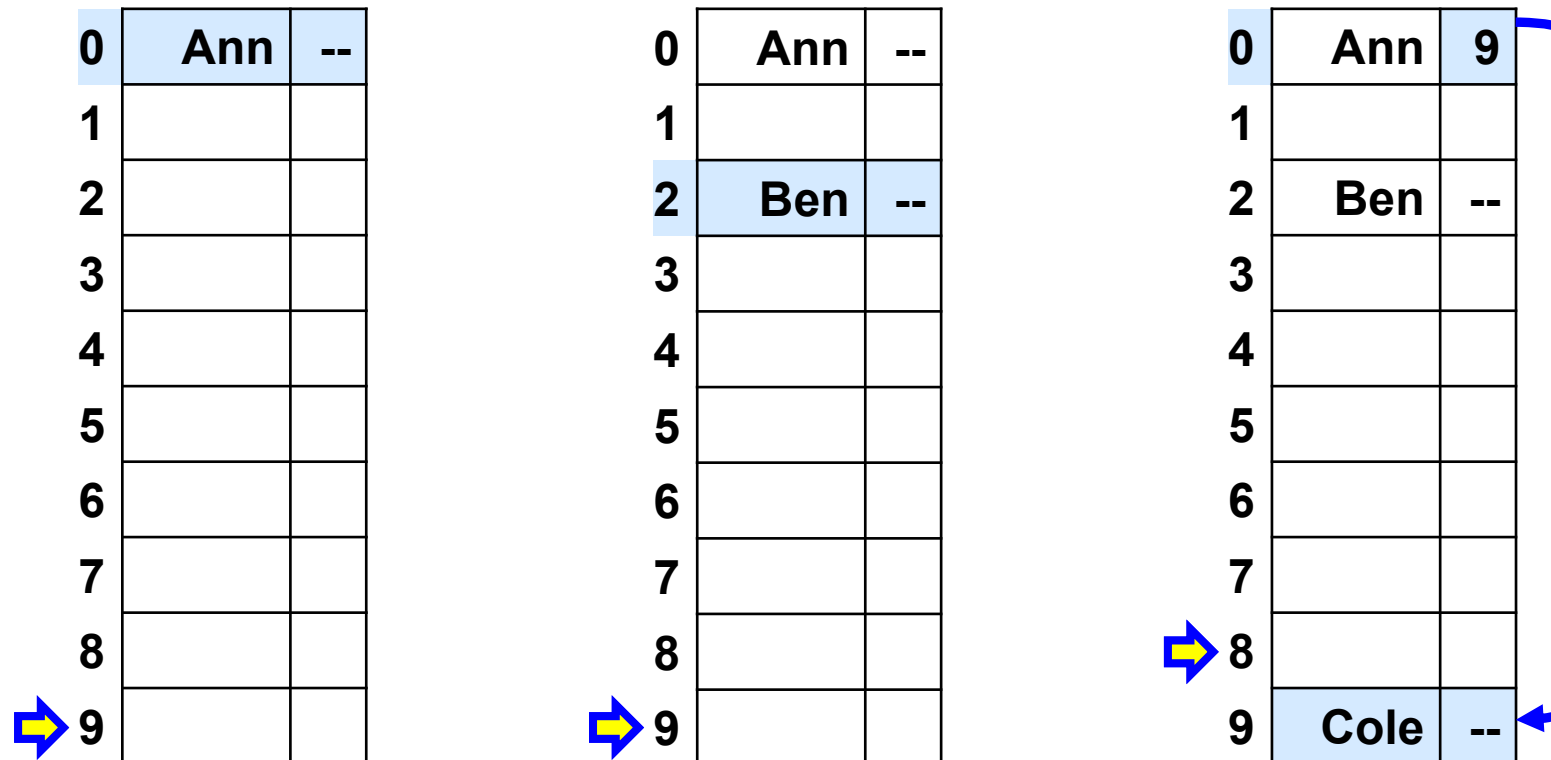
Hashovací funkce  $h$ , data  $d$ .  
 Pozice  $p := h(d)$ ;  
 Prohledej seznam začínající na pozici  $p$  a pokud nenajdeš  $d$ , přidej  $d$  do tabulky na první volné místo od konce tabulky a připoj ho do seznamu synonym  $d$  na poslední místo.

Ukazatel na první volné místo od konce tabulky.  
 Po každém přidání prvku se aktualizuje.

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

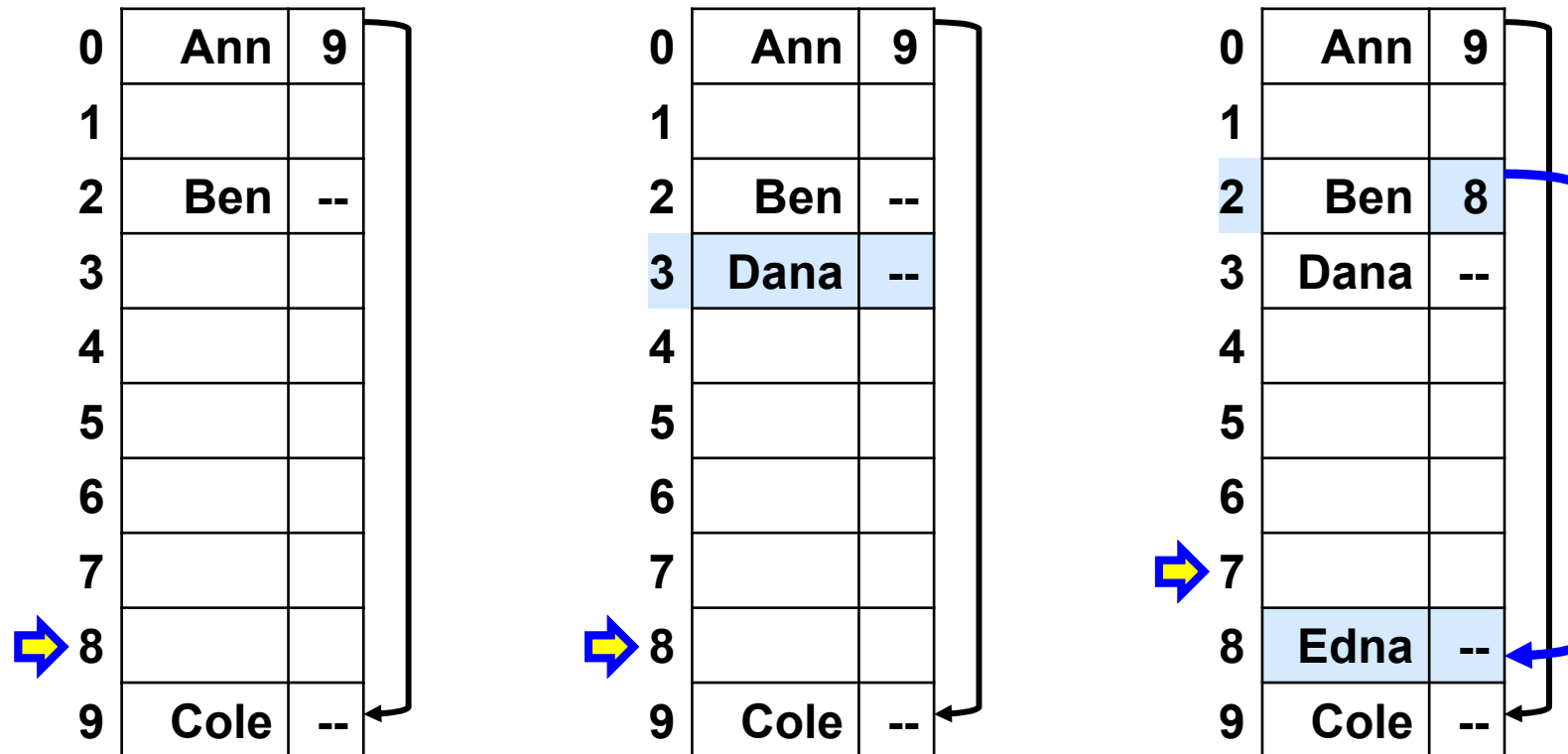
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
$h(\text{data})$	0	2	0	3	2	9	0	8	7

# LISCH (late insert standard coalesced hashing)



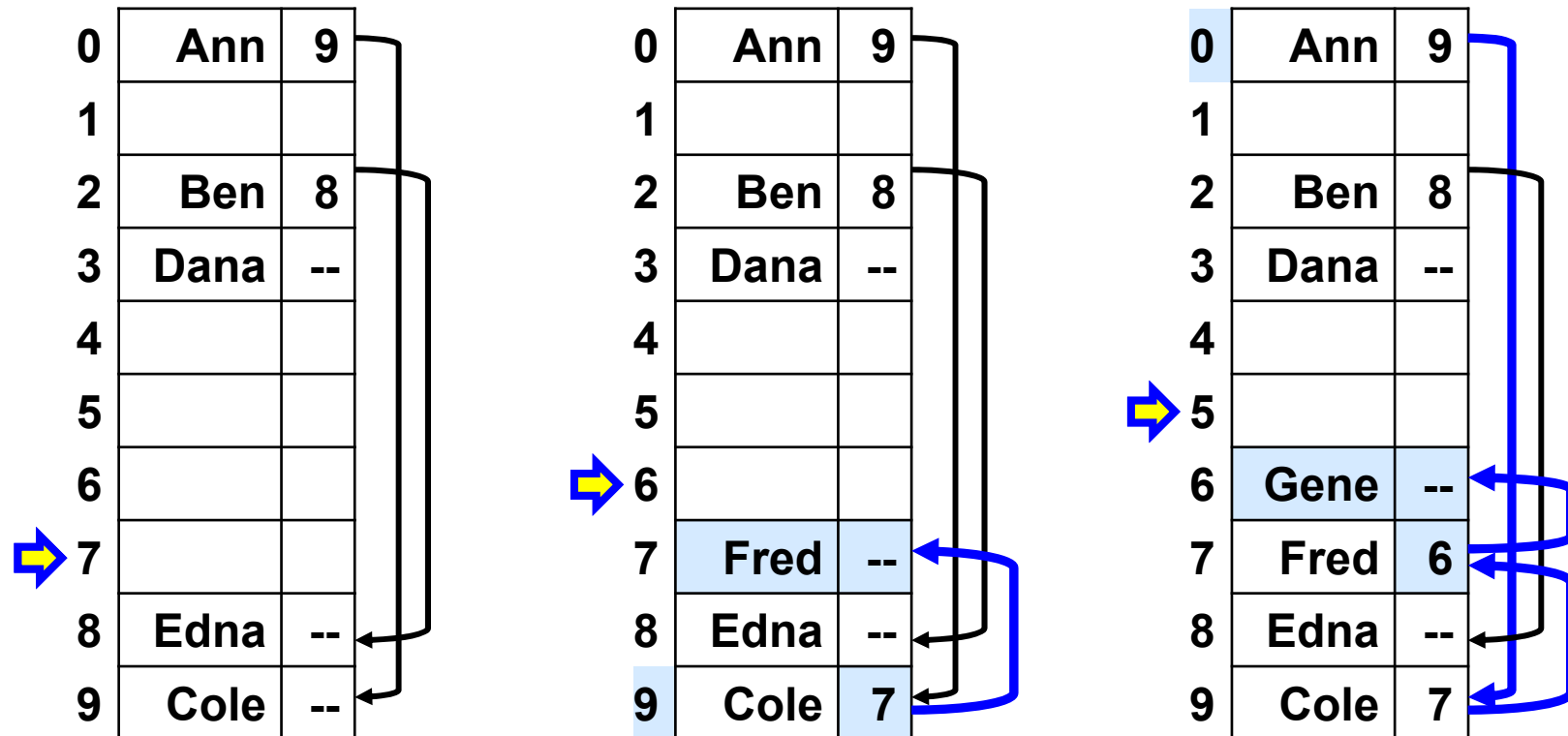
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	3	2	9	0	8	7

# LISCH (late insert standard coalesced hashing)



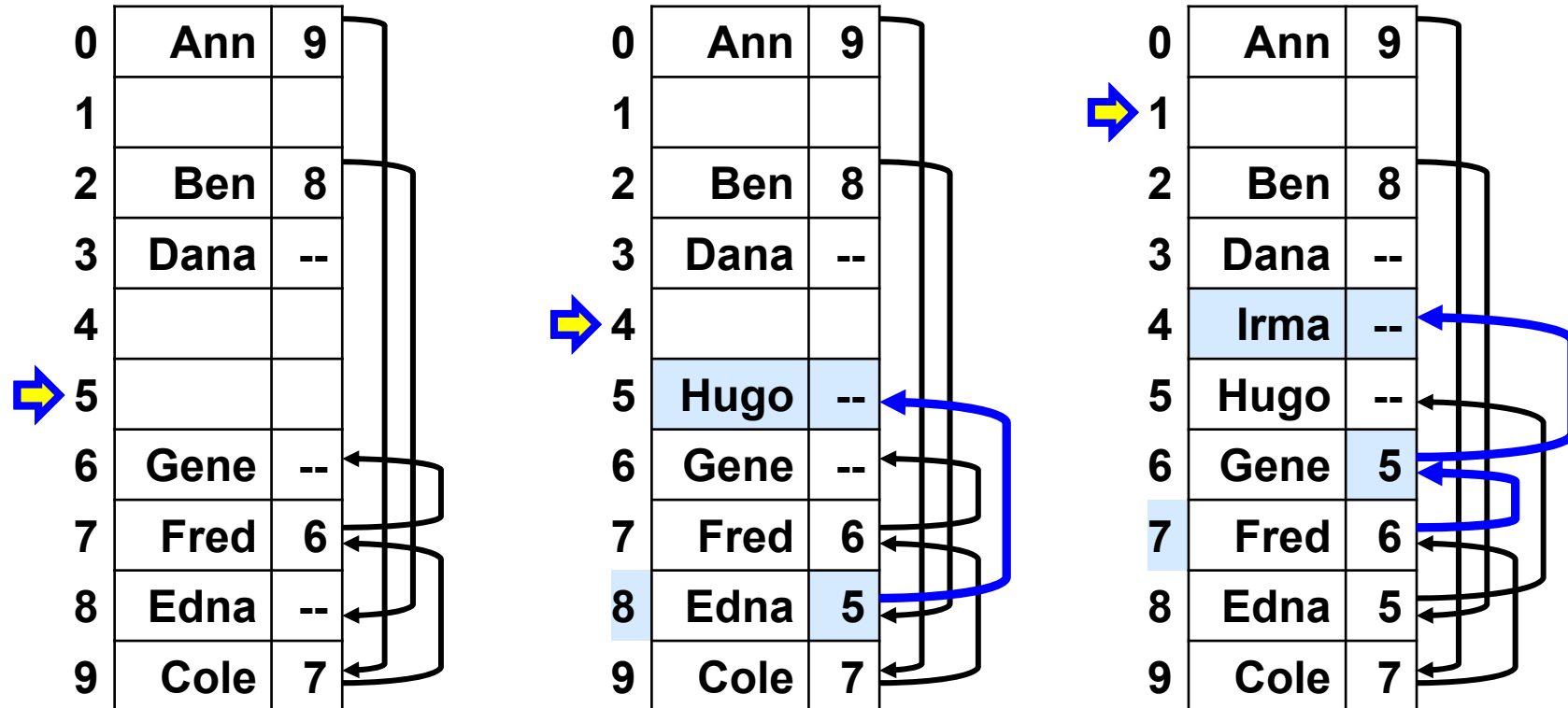
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	3	2	9	0	8	7

# 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)

Hashovací funkce  $h$ , data  $d$ .  
 Pozice  $p := h(d)$ ;  
 Prohledej seznam začínající na pozici  $p$  a pokud nenajdeš  $d$ , přidej  $d$  do tabulky na první volné místo od konce tabulky a připoj ho do seznamu synonym  $d$  za první místo.

Ukazatel na první volné místo od konce tabulky.  
 Po každém přidání prvku se aktualizuje.



	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)

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

→

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

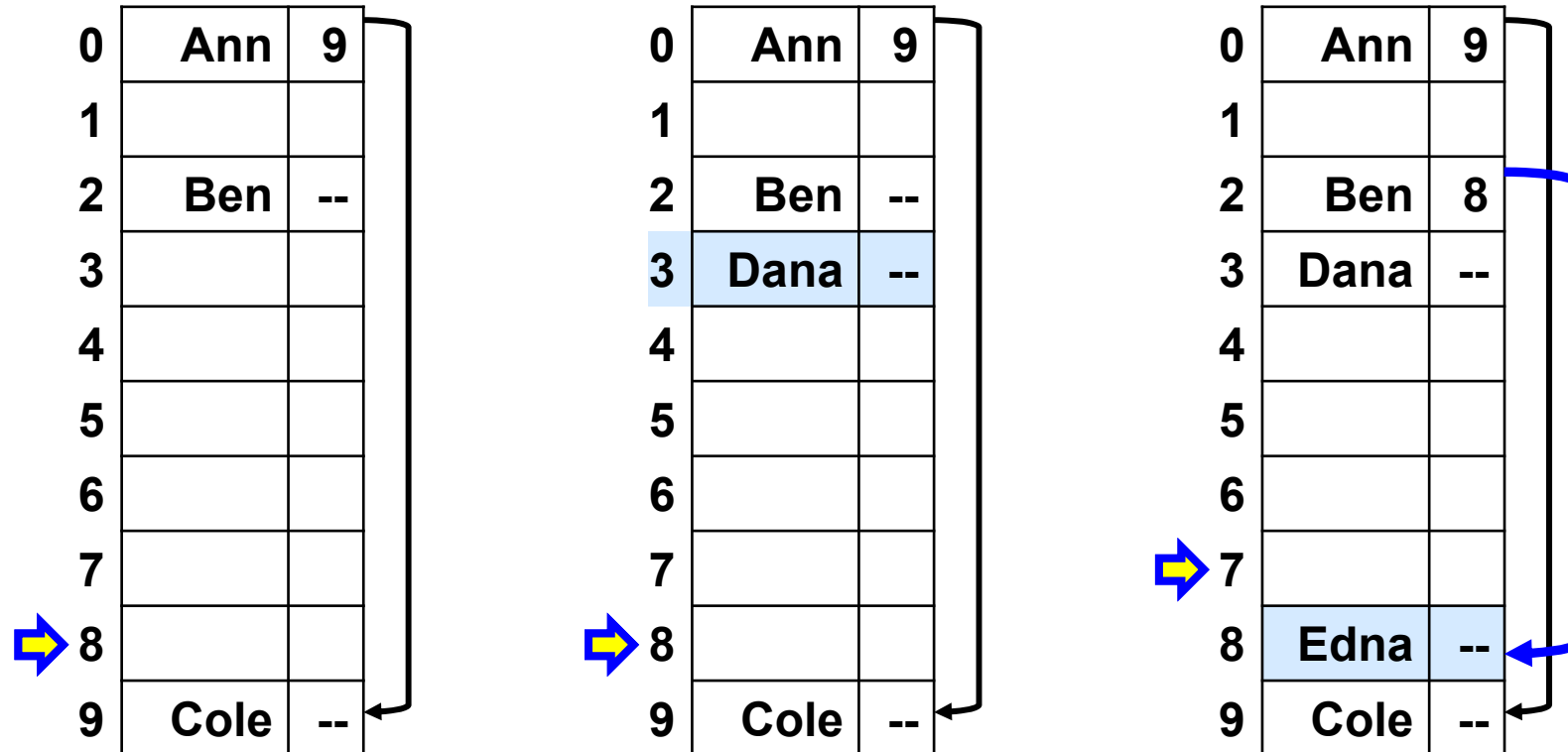
→

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

→

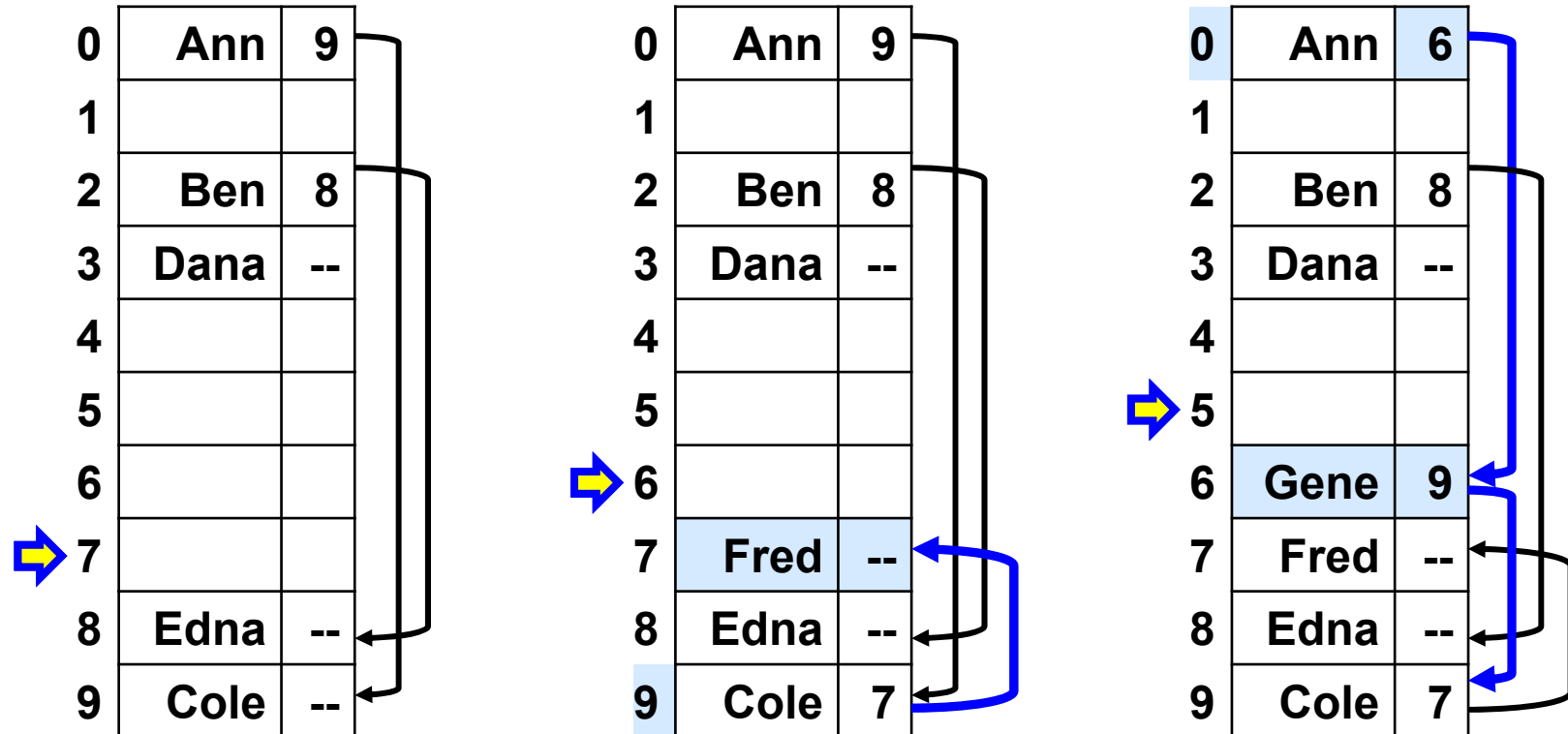
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	3	2	9	0	2	6

## EISCH (early insert standard coalesced hashing)



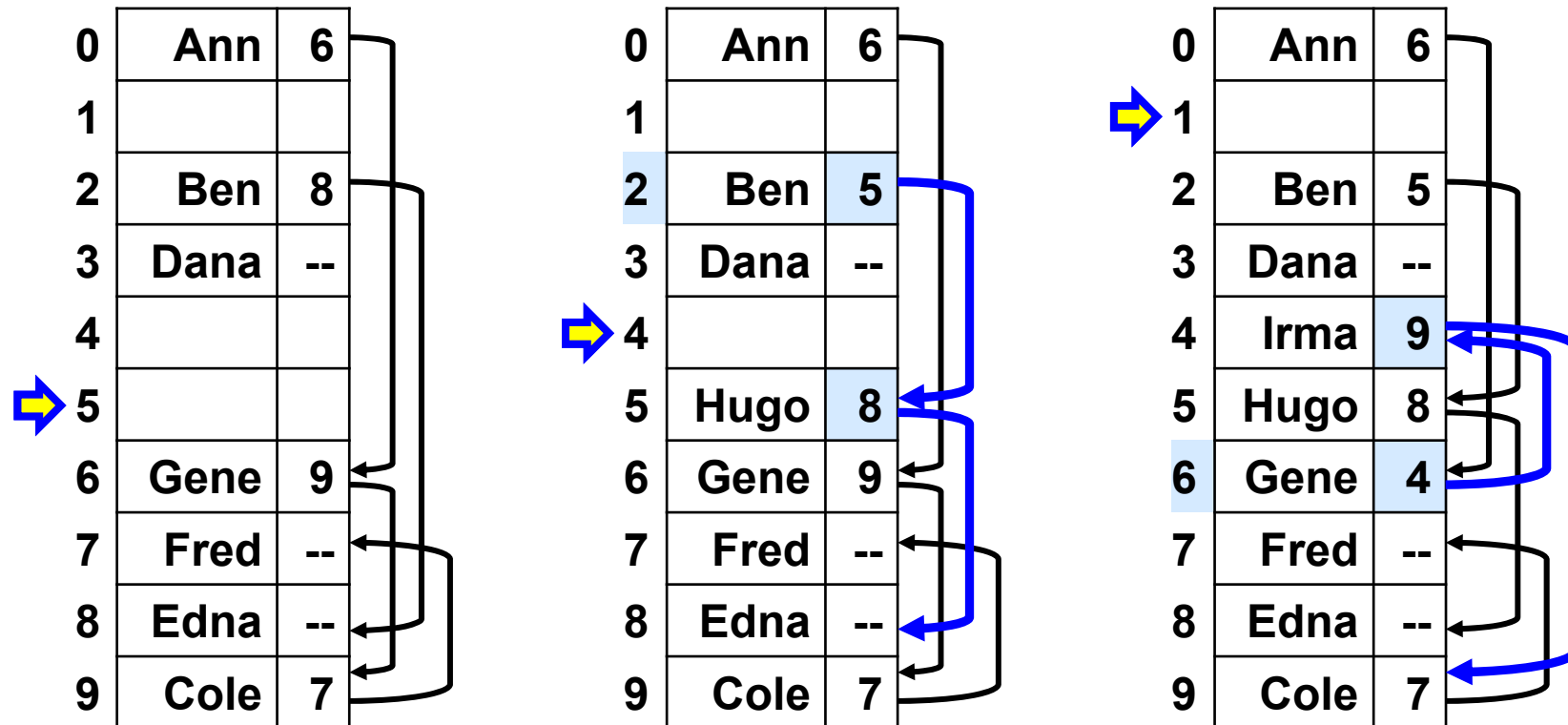
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	3	2	9	0	2	6

# 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

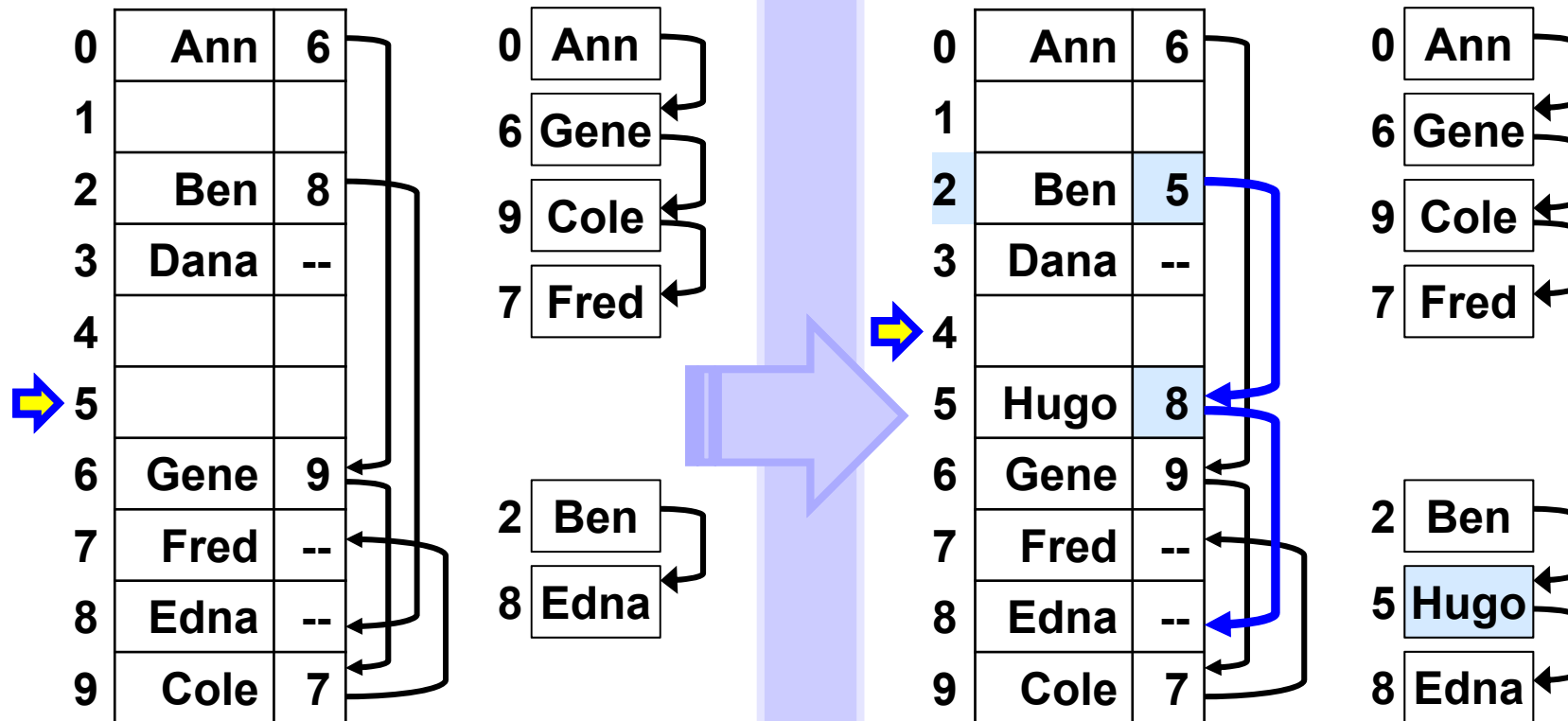
## EISCH (early insert standard coalesced hashing)



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

# EISCH (early insert standard coalesced hashing)

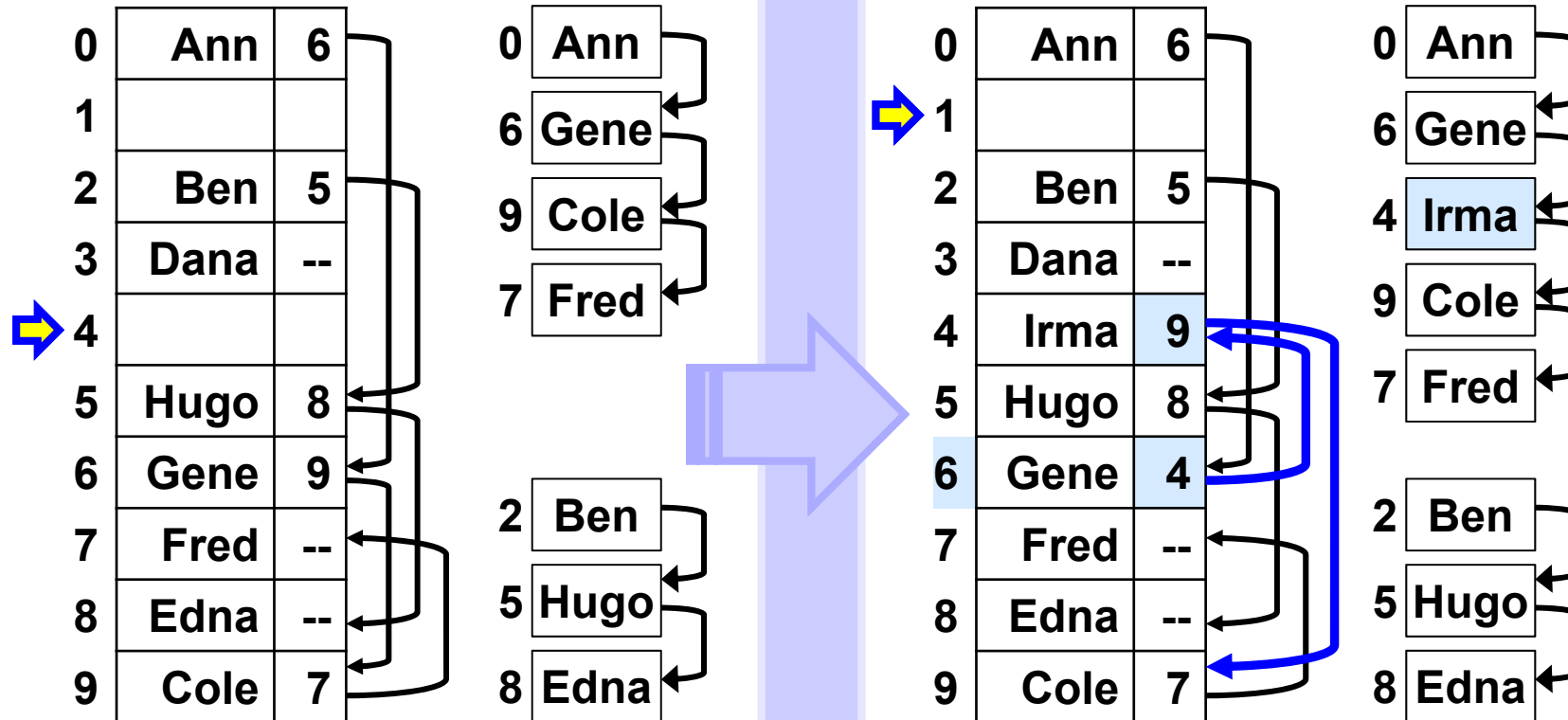
Insert(Hugo)



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

## Srůstající hashování s pomocnou pamětí

**Pro snížení srůstání a tedy zvýšení efektivity hashování se tabulka rozšiřuje o pomocnou paměť - tzv. sklep (cellar).**

**Sklep je místo na konci tabulky, které není adresovatelné hashovací funkcí, má ale stejnou strukturu jako celá tabulka.**

**Algoritmy LICH a EICH jsou analogické varianty algoritmů LISCH a EISCH s přidáním sklepa.**

**Po naplnění sklepa pokračuje plnění jako v LISCH a EISCH.**

**Algoritmus VICH (variable insert coalesced hashing) připojuje prvek za poslední prvek seznamu, který je ještě ve sklepe. Pokud ve sklepe žádný není, vkládá jako EISCH, tj. hned za kolidující prvek v seznamu.**

## LICH (late insert coalesced hashing)

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

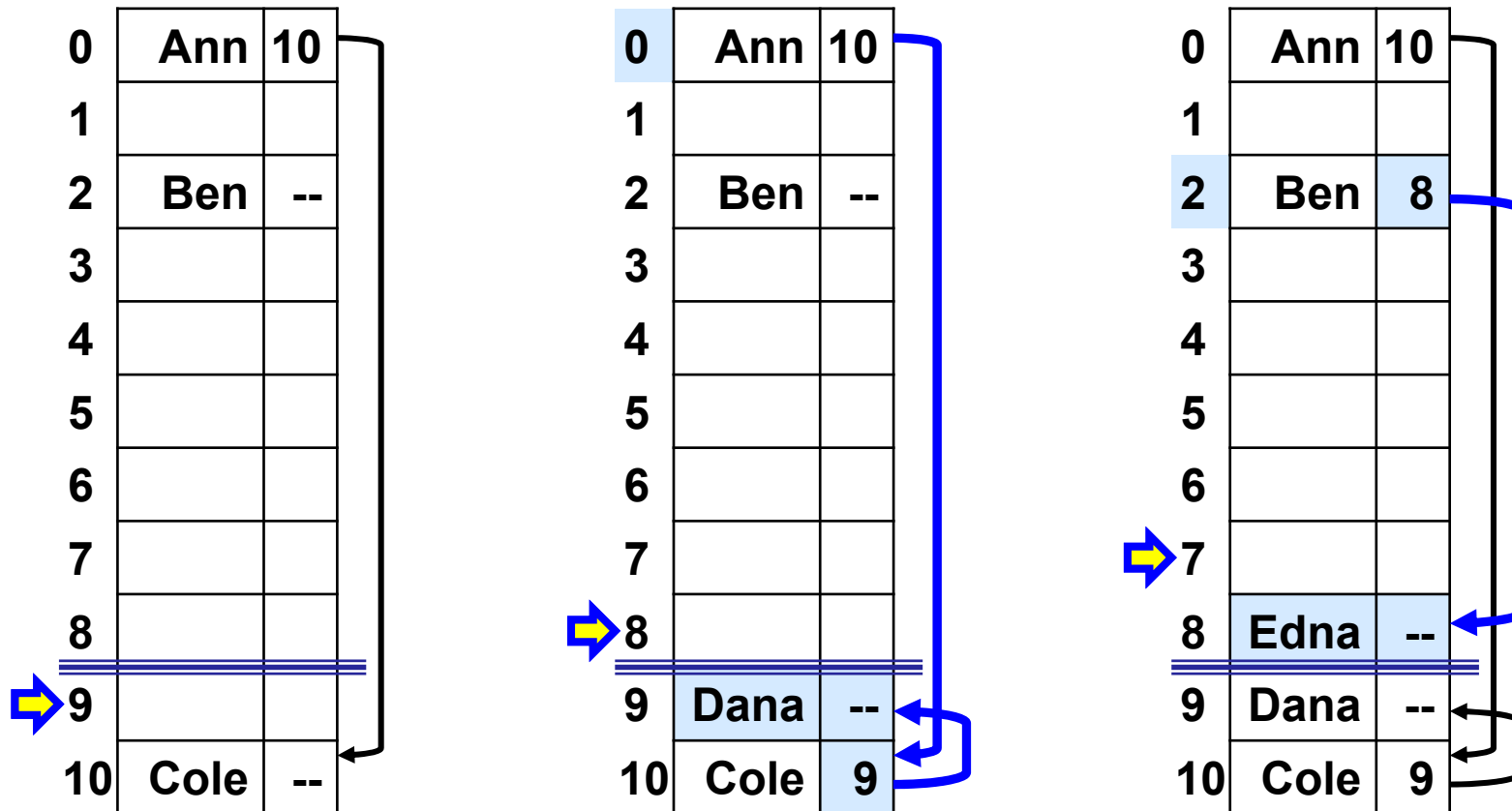
0	Ann	--
1		
2	Ben	--
3		
4		
5		
6		
7		
8		
9		
10		

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

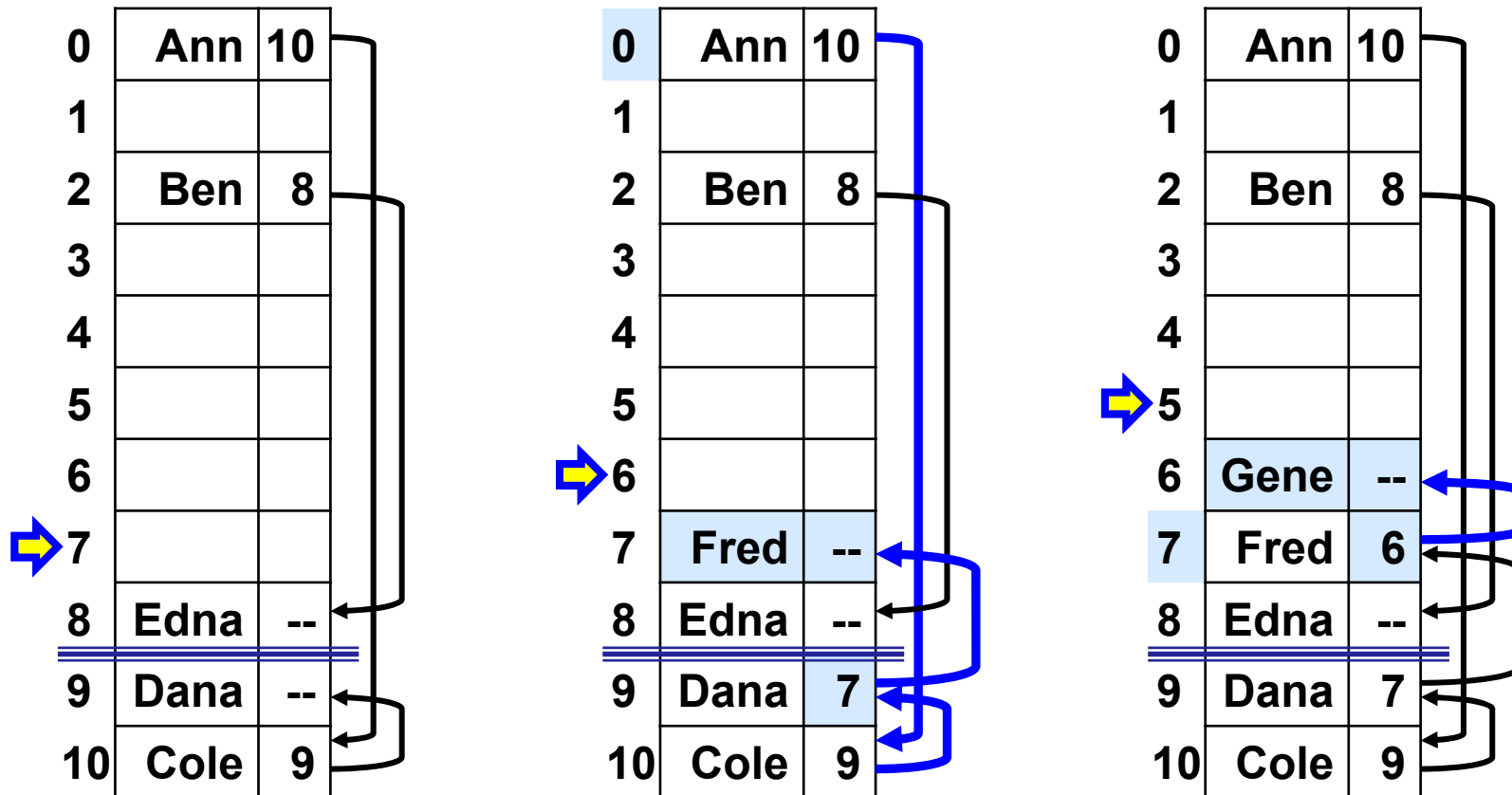


## LICH (late insert coalesced hashing)



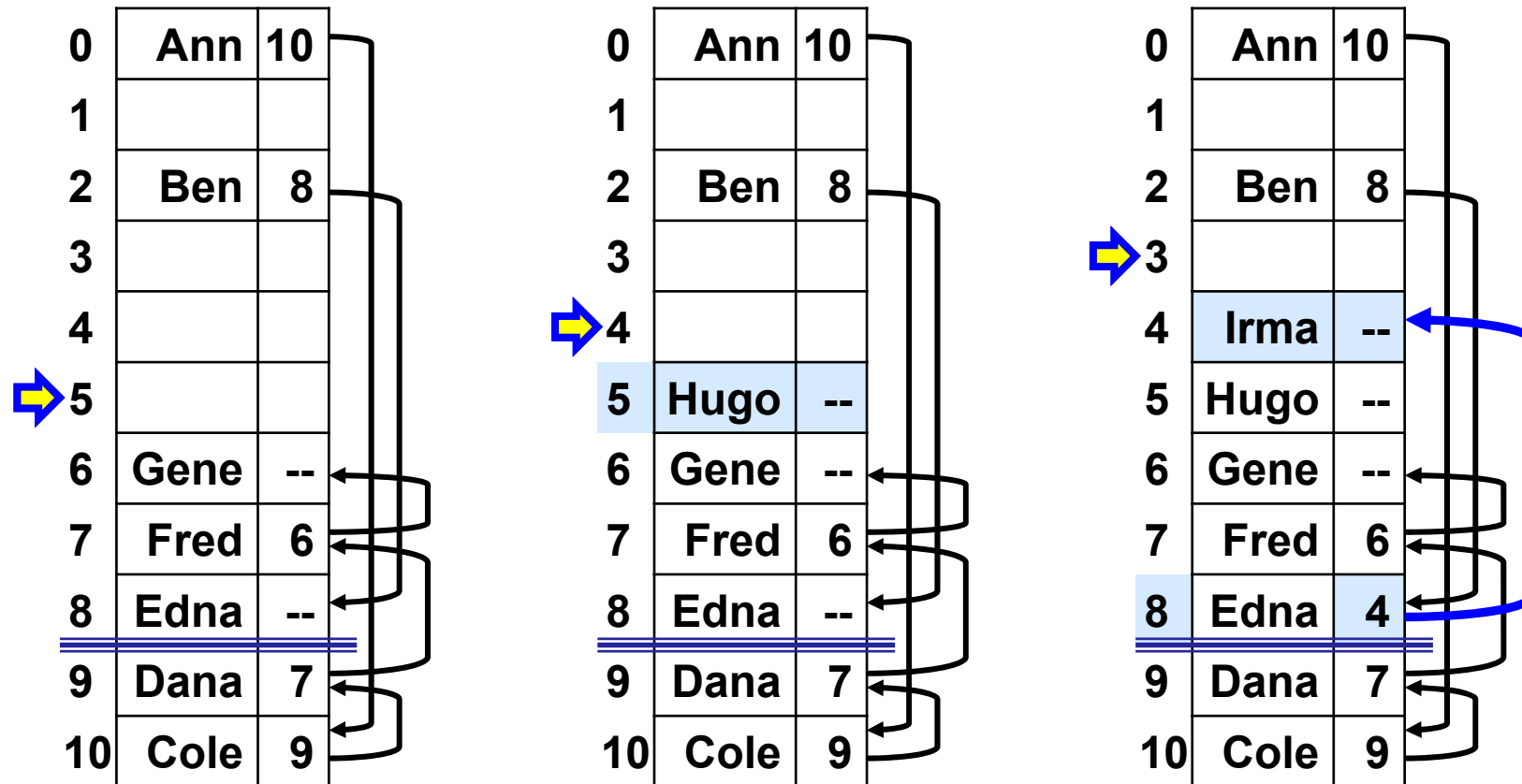
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	7	5	8

## LICH (late insert coalesced hashing)



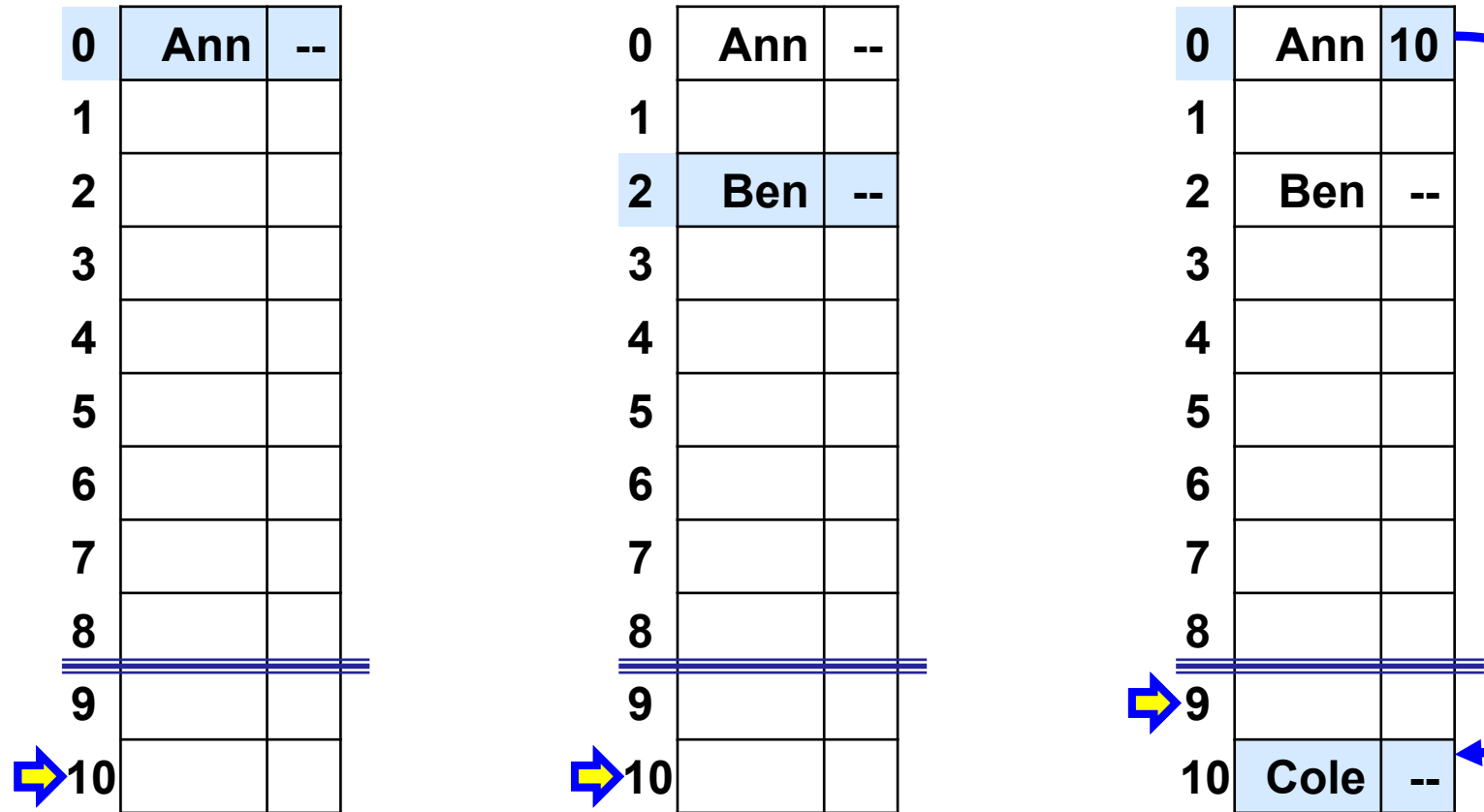
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	7	5	8

## LICH (late insert coalesced hashing)



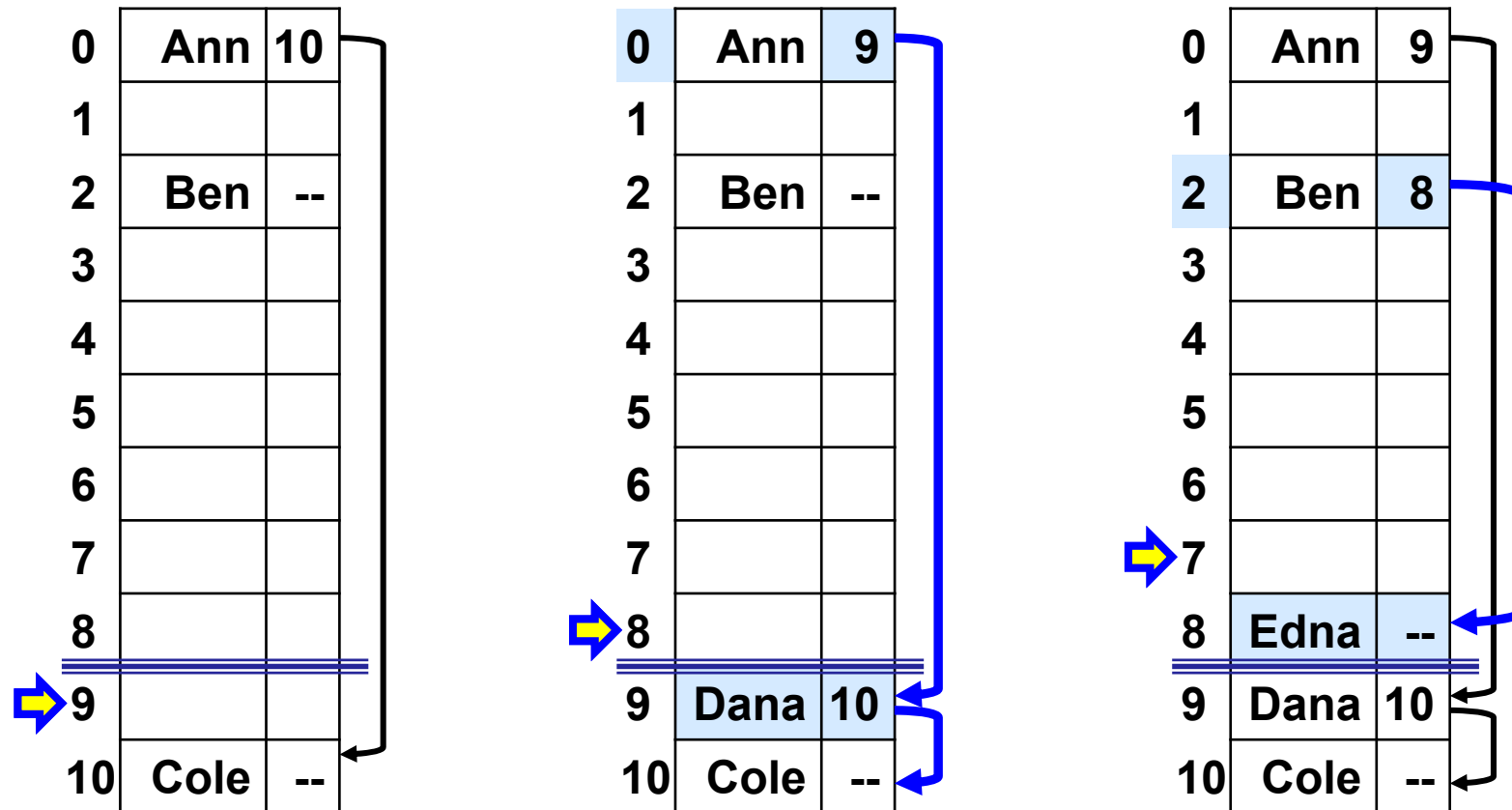
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	7	5	8

## EICH (early insert coalesced hashing)



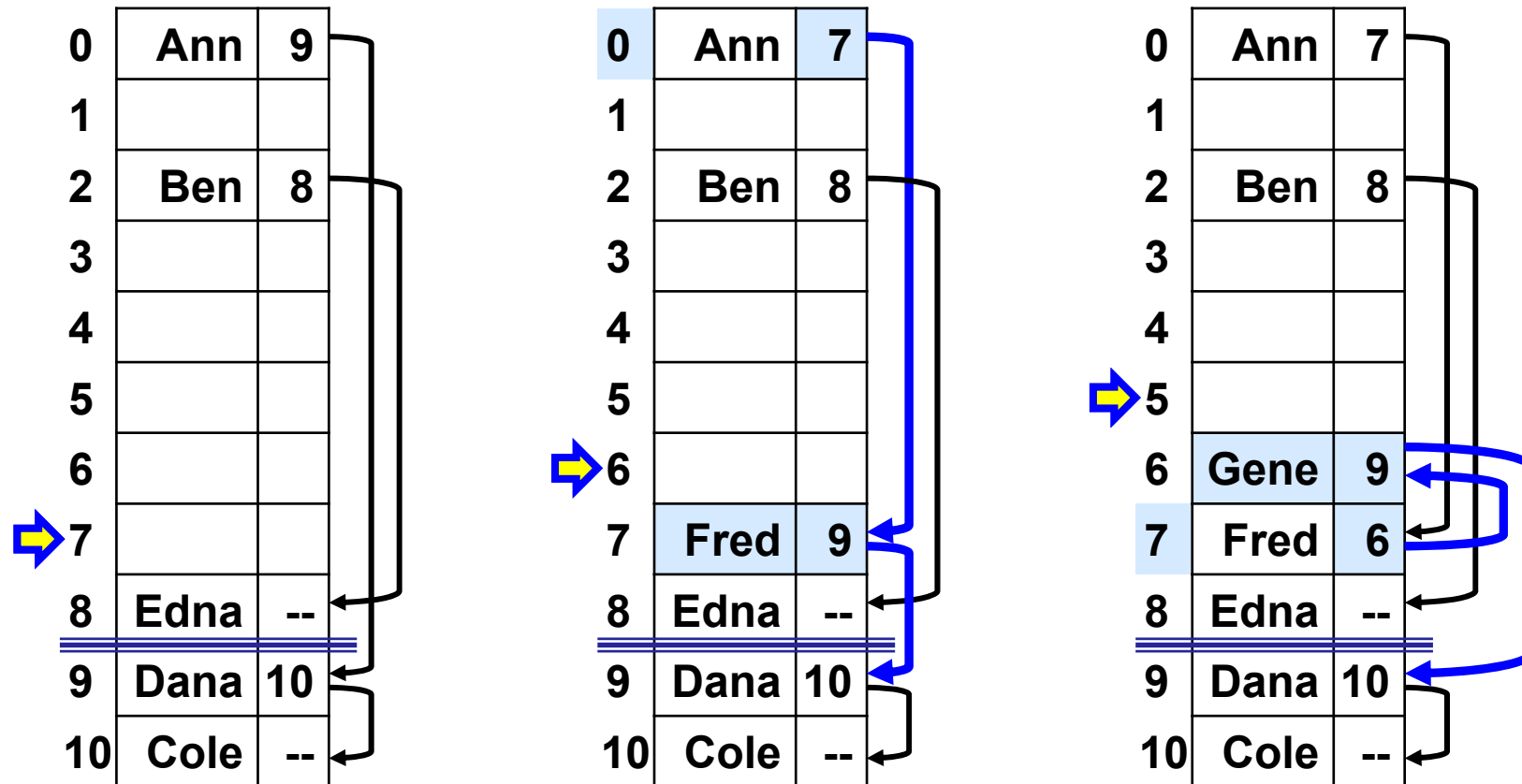
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	7	0	8

## EICH (early insert coalesced hashing)



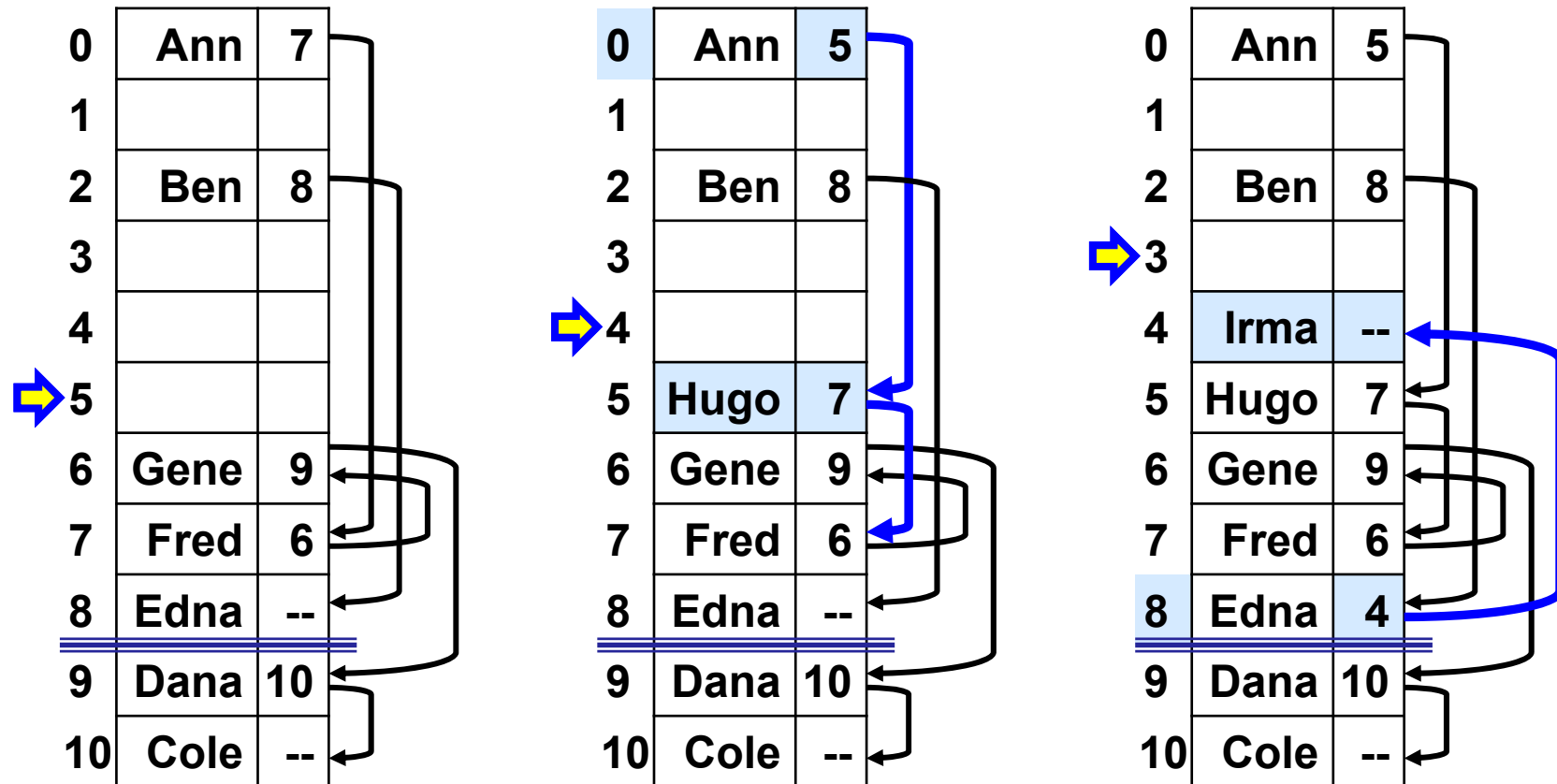
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	7	0	8

## EICH (early insert coalesced hashing)



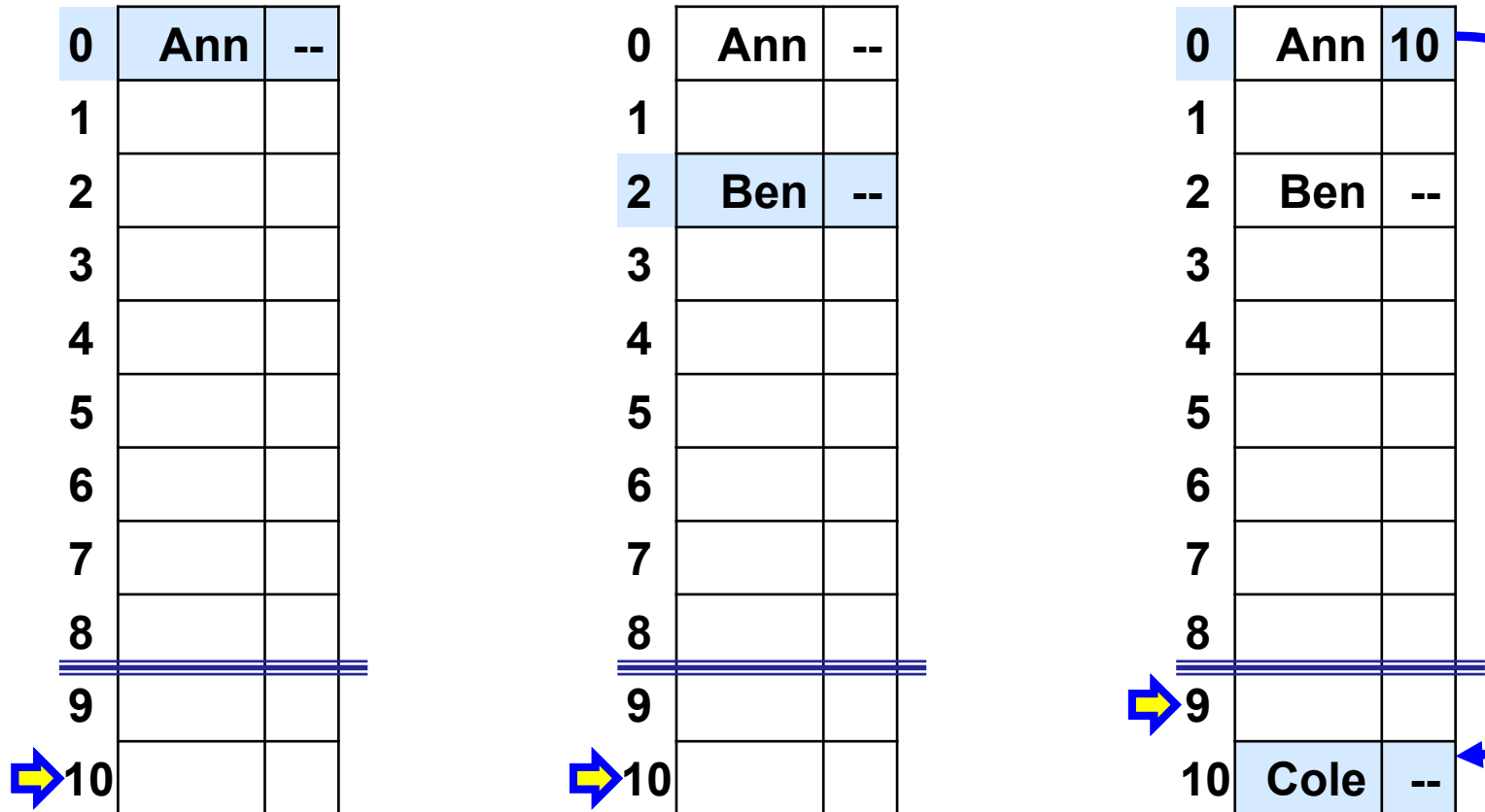
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
$h(\text{data})$	0	2	0	0	2	0	7	0	8

## 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

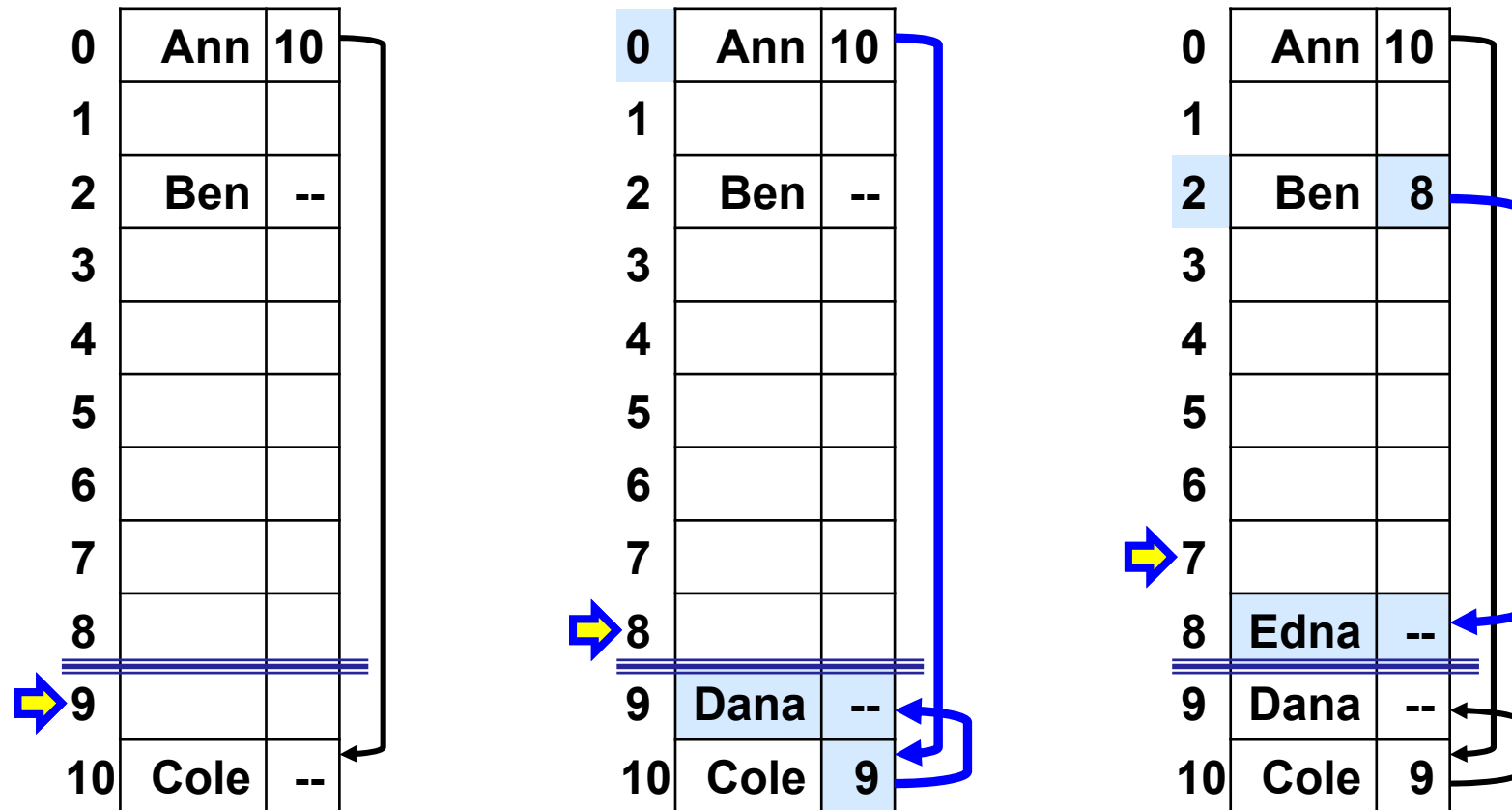
## VICH (variable insert coalesced hashing)



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

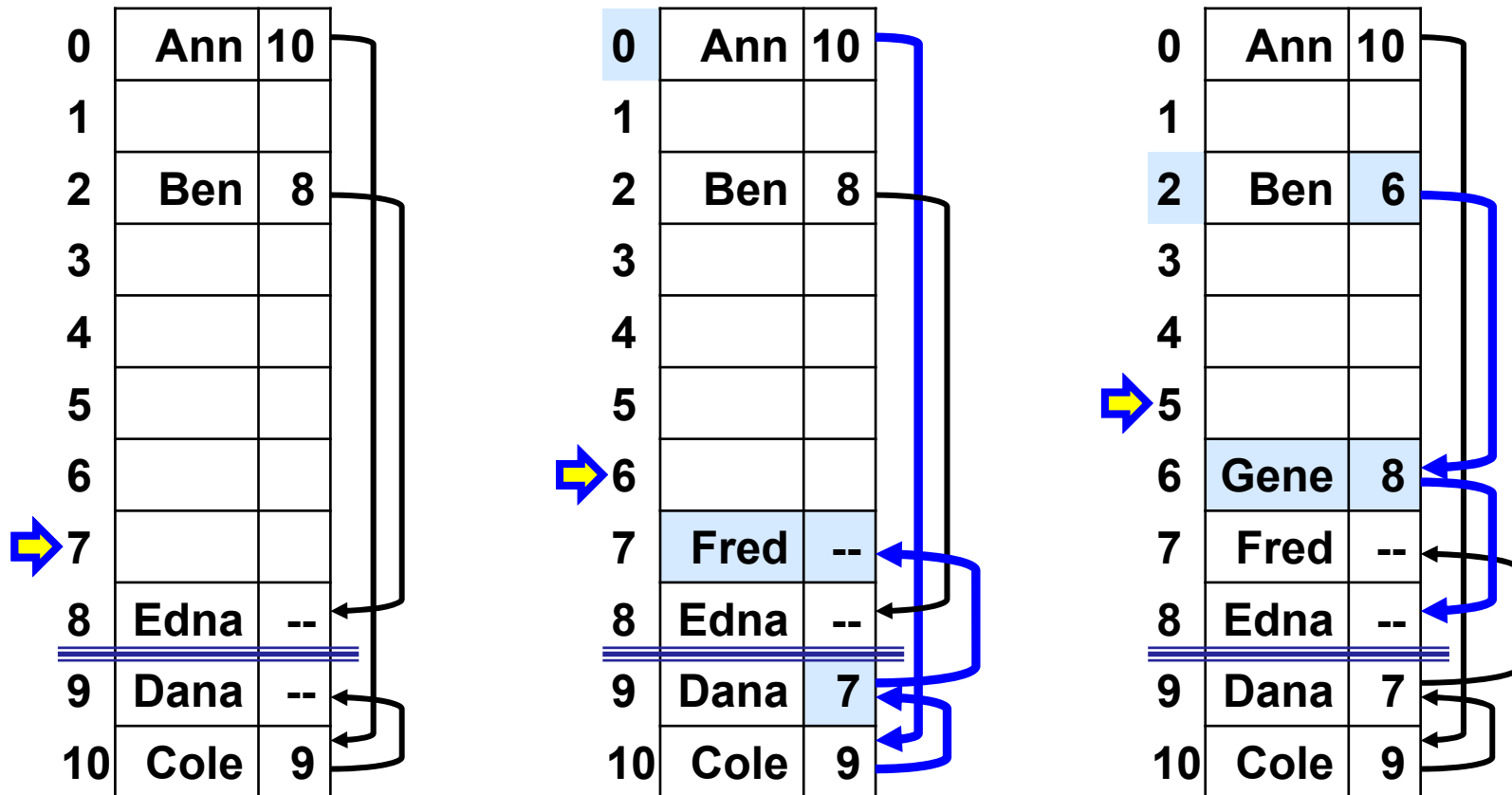


## VICH (variable insert coalesced hashing)



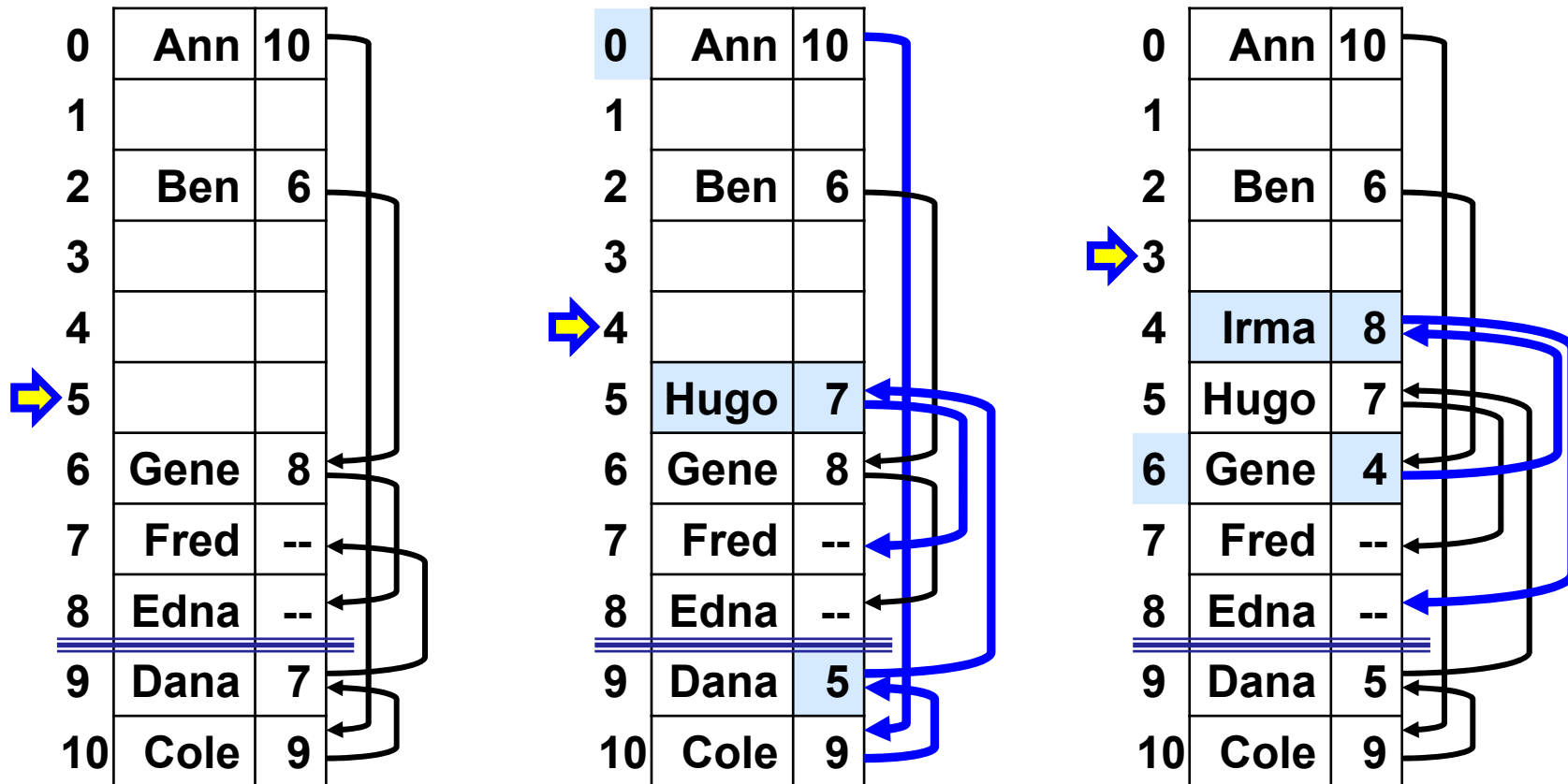
data	Ann	Ben	Cole	Dana	Edna	Fred	Gene	Hugo	Irma
h(data)	0	2	0	0	2	0	2	0	6

## 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