D = 3 — Factor = 2

0	000	0
1	001	1
→ 2	010	-1
3	011	-1
4	100	-1
5	101	-1
6	110	-1
7	111	-1

0 | a, c

1 | b, d

1 | b, d, e

d

01 | e

11 | b, d

01 → a 000 → c

01 1 → b 111 → d 101 → e

D = 3

0	000	0
1	001	1
→ 2	010	-1
3	011	2
4	100	-1
5	101	-1
6	110	-1
7	111	3

d=1

0 | a, c

d=3

01 | e, m

d=3

101 | a

011 | b, d

d=1

111 | x, r

d=2

1 1

1

001

1 01

111

d=1

0 | a, c

a, c

0

e

01

11

b, d

011

111

X

111 | b, d, x

d=3

011 | b, d

d=2

111 | x

001

1 01

111

1 01

Insert 010 → 0

↳ c

0	000	0
1	001	1
→ 2	010	-1
3	011	-1
4	100	-1
5	101	-1
6	110	-1
7	111	-1

d=1

0 | a, c

d=1

1 | b

insert 010

if (a == -1)

↳ 010 && 1 = 0

0 | 000 | -1

IF (==-1)

Create bucket

110

40%d 8

010

010

010

↳ a

d=1

Insert 010 → d

↳ d

0	000	0
1	001	1
→ 2	010	-1
3	011	-1
4	100	-1
5	101	-1
6	110	-1
7	111	-1

d=1

0 | a, c, d

d=1

1 | b

Split

00 | a, c

10 | d

mask = 1

Insert 010 → 0

↳ c

0	000	0
1	001	1
→ 2	010	2
3	011	-1
4	100	-1
5	101	-1
6	110	-1
7	111	-1

d=2

00 | a, c

d=1

1 | b

d=2

10 | d

100

Insert 110 → 0 d=2 ⇒ 10 Invertay

↳ e

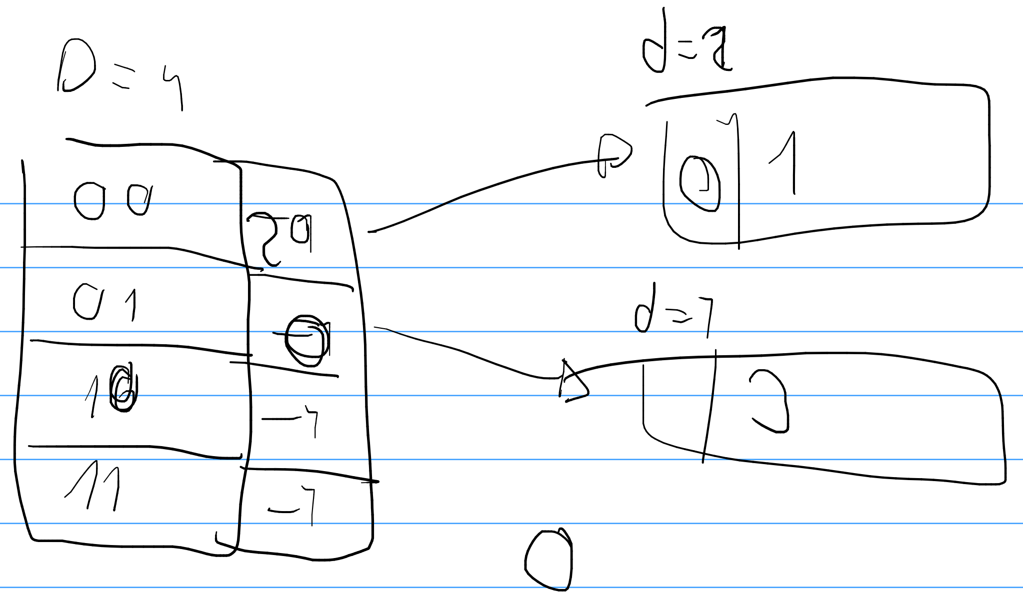
000

110

d = 2 mask = 11

t = 2 mask = 11

t = 1 11 ∨ && 11 = 10



2

0

10

10

for(Buckets)

a b c

00 10 10

a, b, c

Split (bucket, ~~index~~ mask, t)

t = 2

11

bucket

00

10

Mask = 0

t = 1

10

0

d = 1

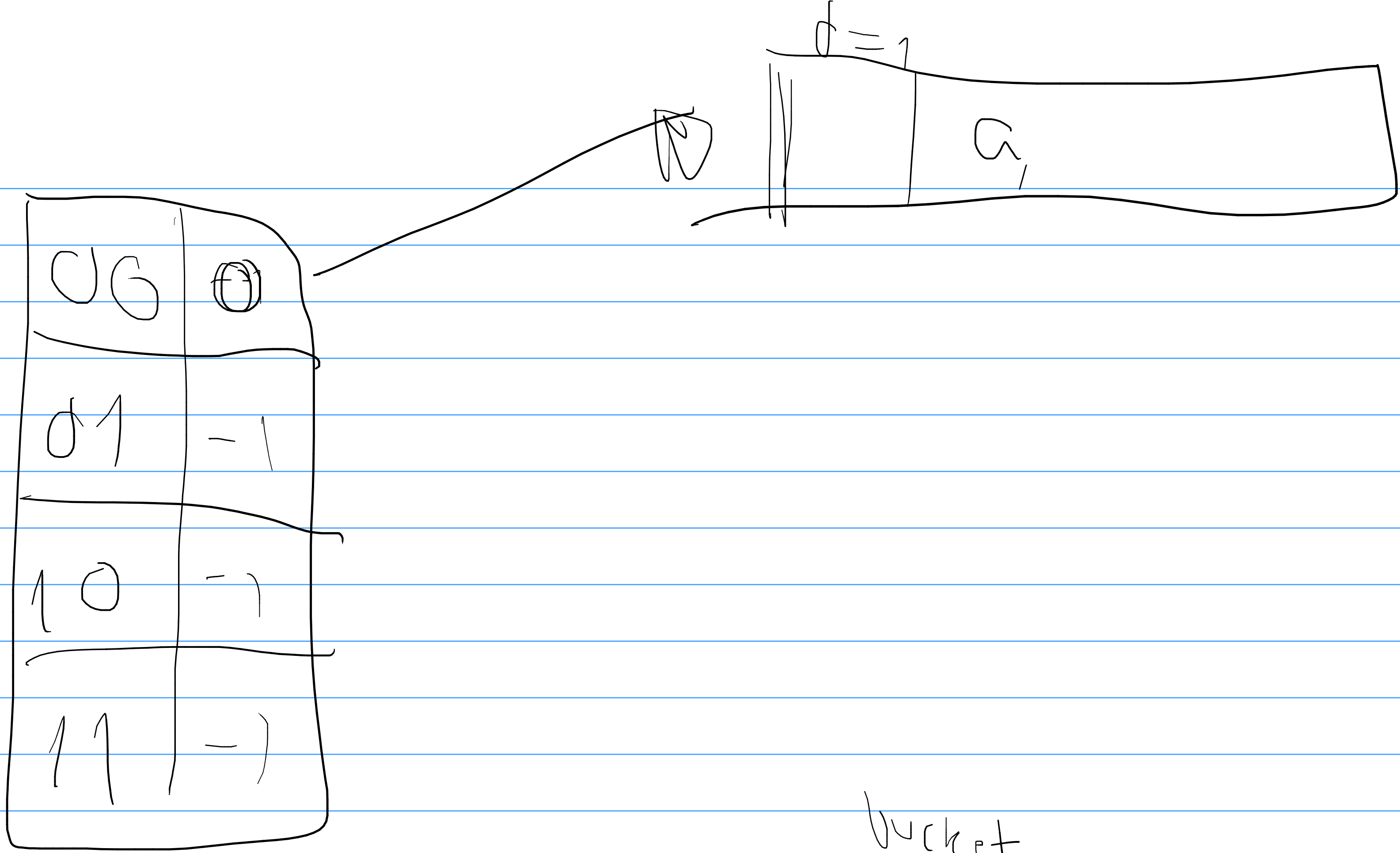
t = 1

00

10

00

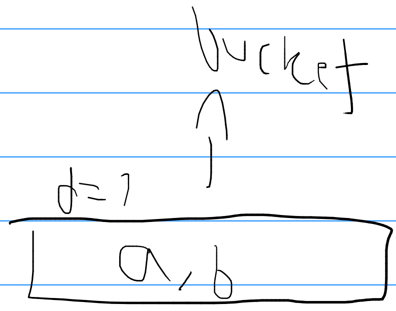
mask



00 → a → 0

10 → b → 0

Abrimos



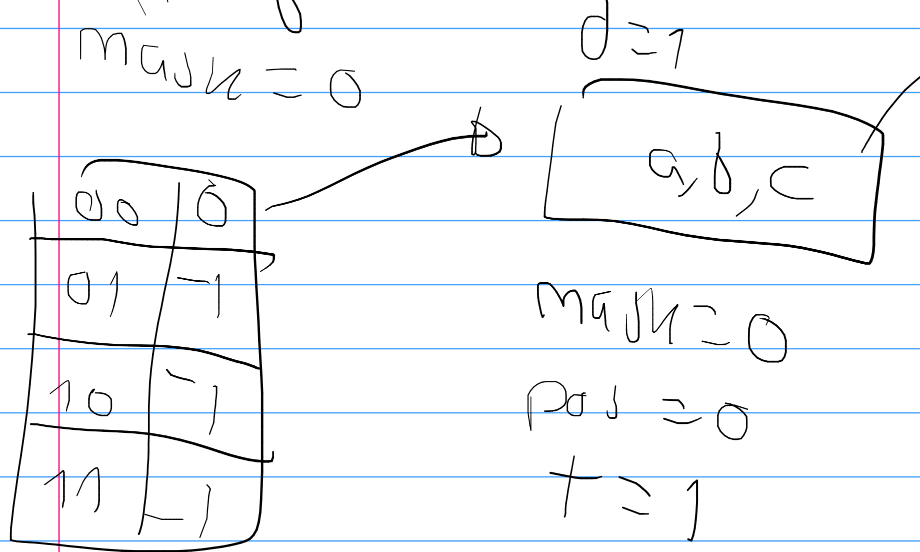
int pos = mask = 0

t = 1

key = b

index = b

mask = 0



→ lleno

new_index = 0 | 10
= 10

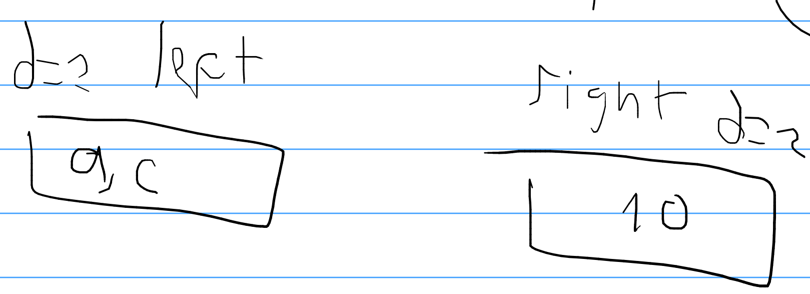
new_mask = 1 | 10 = 11

mask = 0

pos = 0

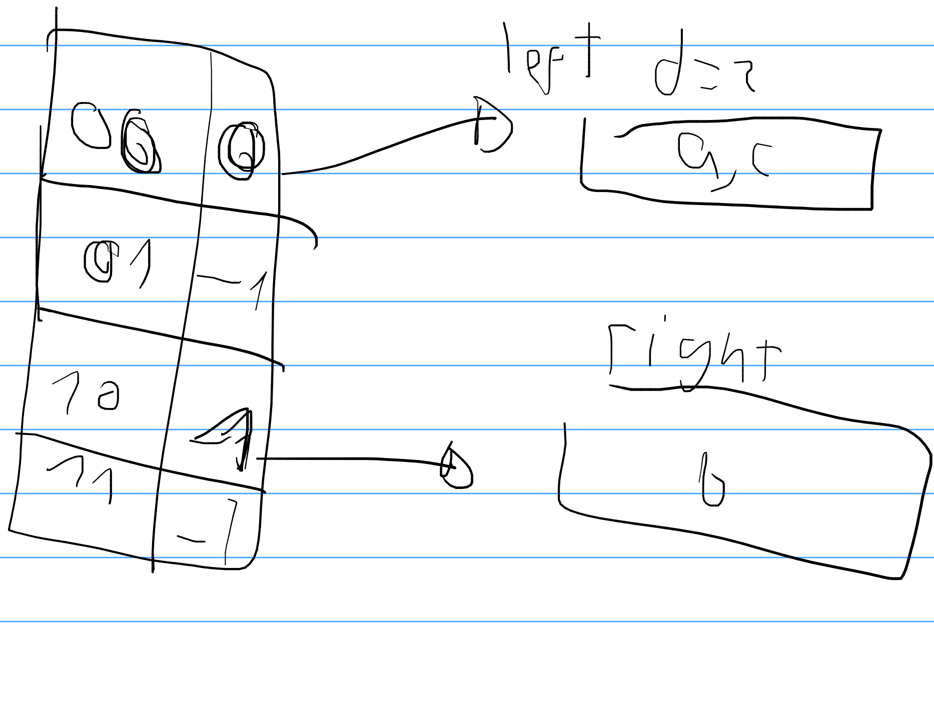
t = 1

mask2 = 1



	value
a → 00	00
b → 10	10
c → 00	00

⇒

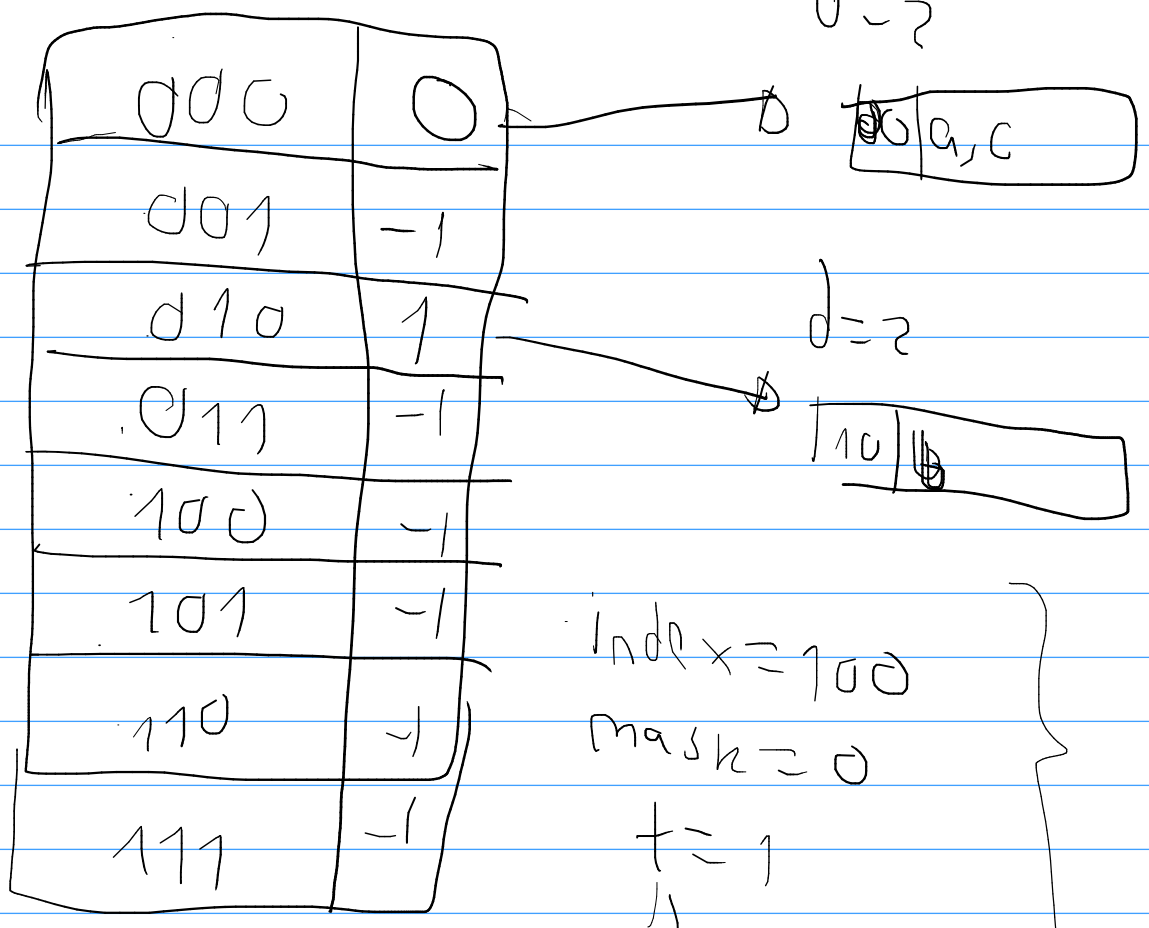


d=1

1 | 10 → 11

d=2

000



d=2

a,c

d=2

10

Insut 100

index=100

mask=0

t=1

d=2

pos=0

mask=1

while( )

1ª Iteración

mask=11

mask=100 & 11

mask=00

Pos=0

d=2

t=2

sale del bucle

a,c,b

Split

d

a=110

b=010

c=010

d=2

a,b,c

010

010

010 | 100 = 110

2

new_index=010 | 100 = 110

new_index=00 | 100 = 100

new_mask=111

left d=3

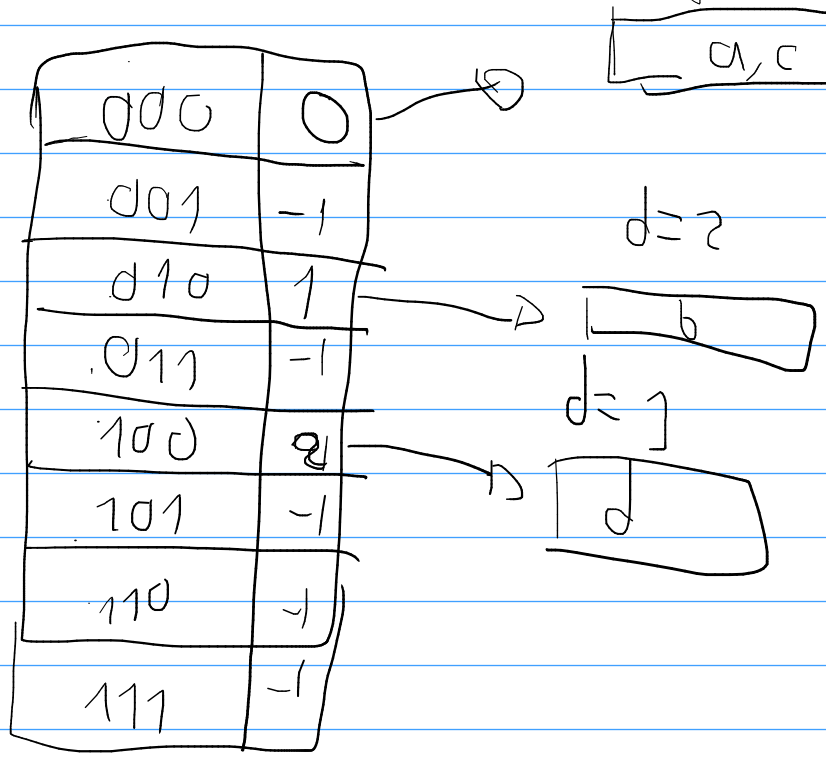
a,c

Right d=3

b

left d=2

a,c

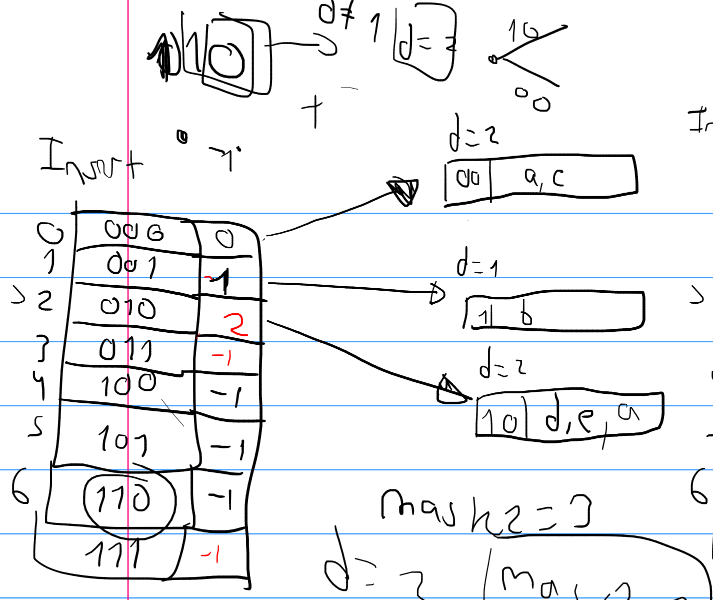


d=2

b

d=1

d

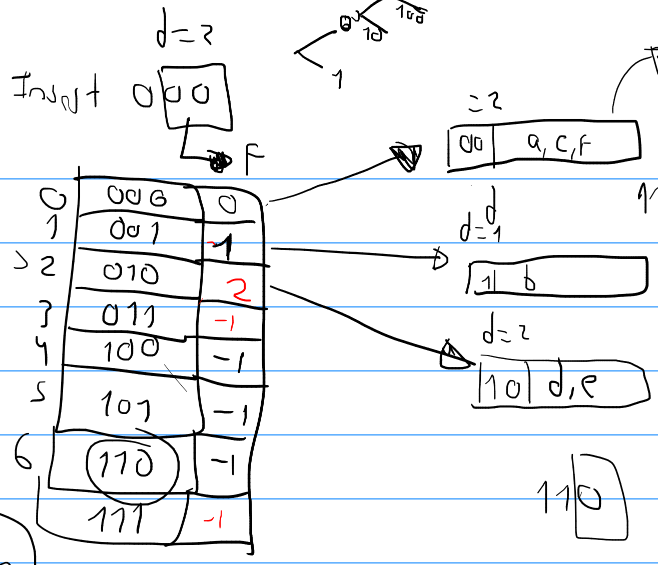


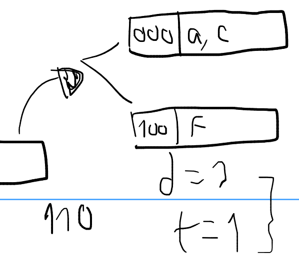
d=2 (mask2=3)

100

110

t=1 (t=2)





110

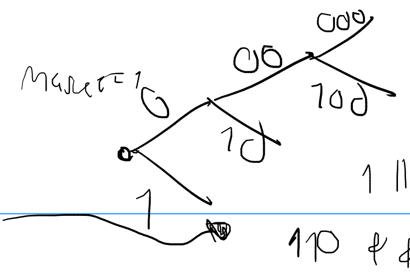
110

Mask 110 & 11 = 10

11 || 100

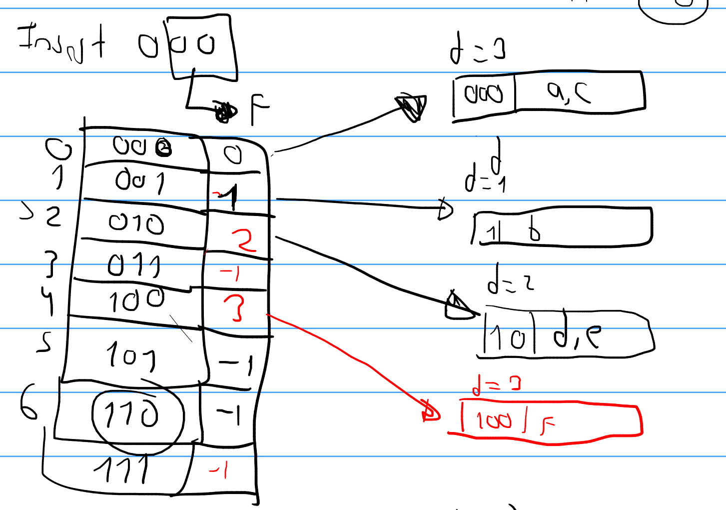
2²+2¹+1

mask2=3



1 || 10 = 11

110 && 11 = 10

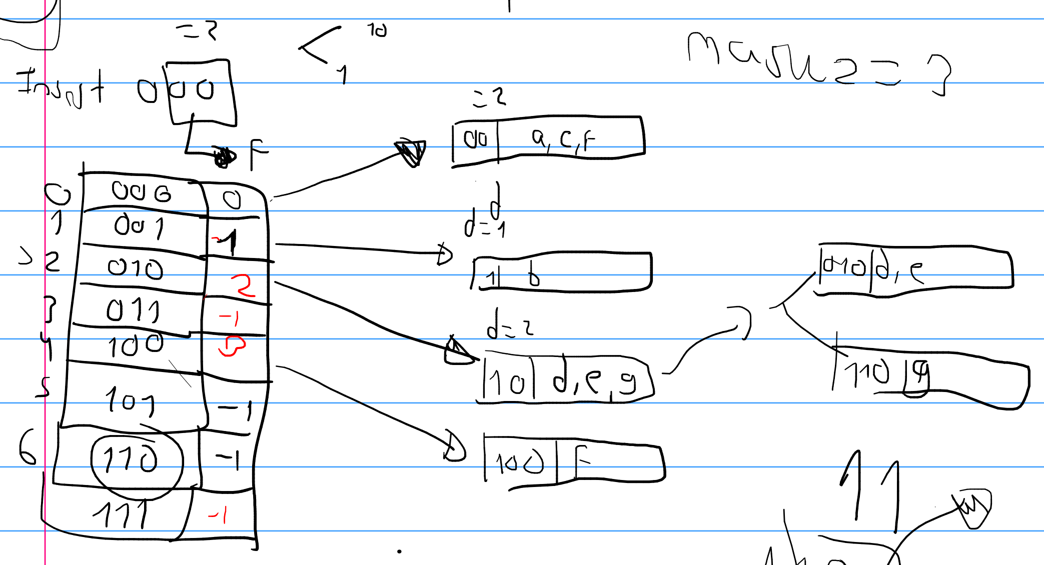


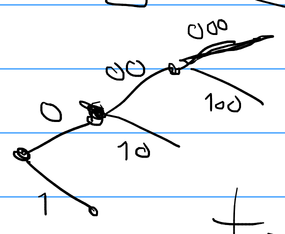
110 → 000 d≠1

t=1 t≠d

t=1

d=2





while ( ) t=2

110

110 && 11 = 10

t=2

1 | 100 = 7

t=1 d=1

011 t=d

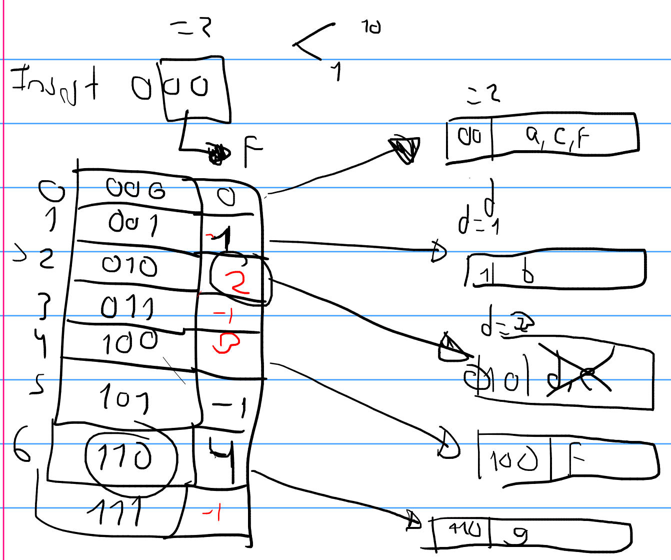
Mask2=3

11

110

Input 110

mask2=11



Parada del arbol

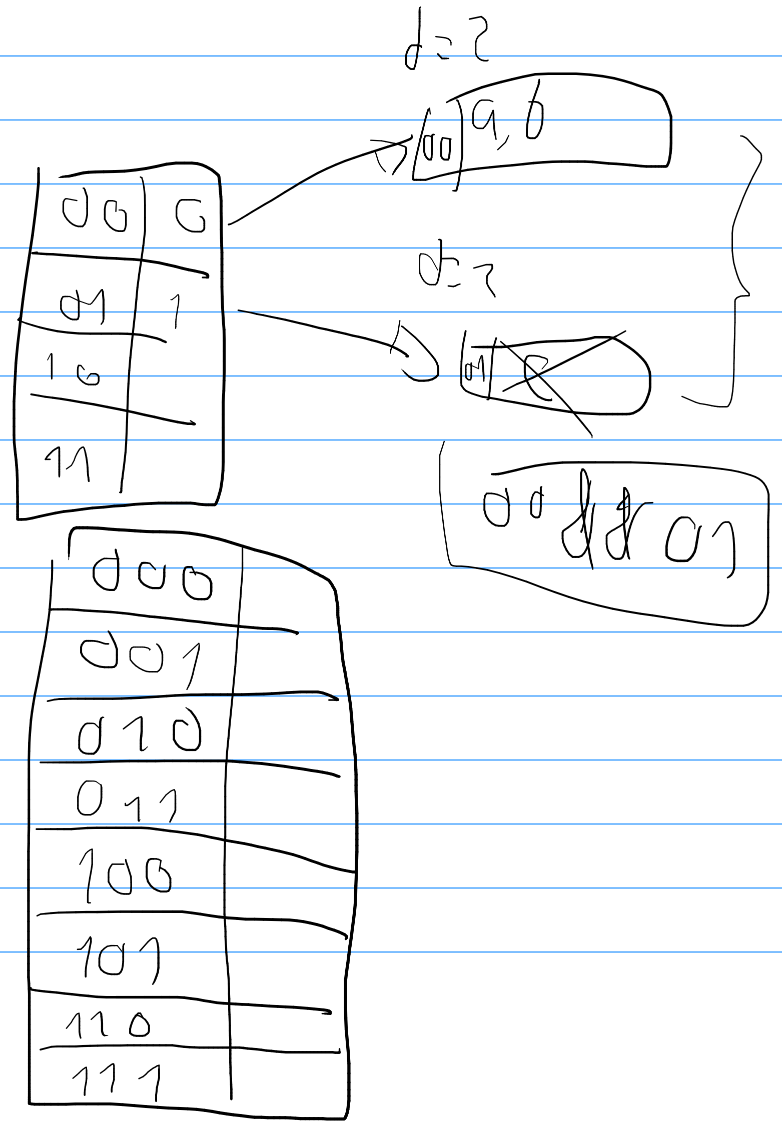
t==d

Merge

d=1 d=1

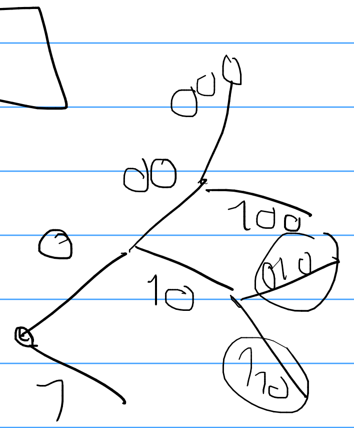
010 && 110

010 d=2

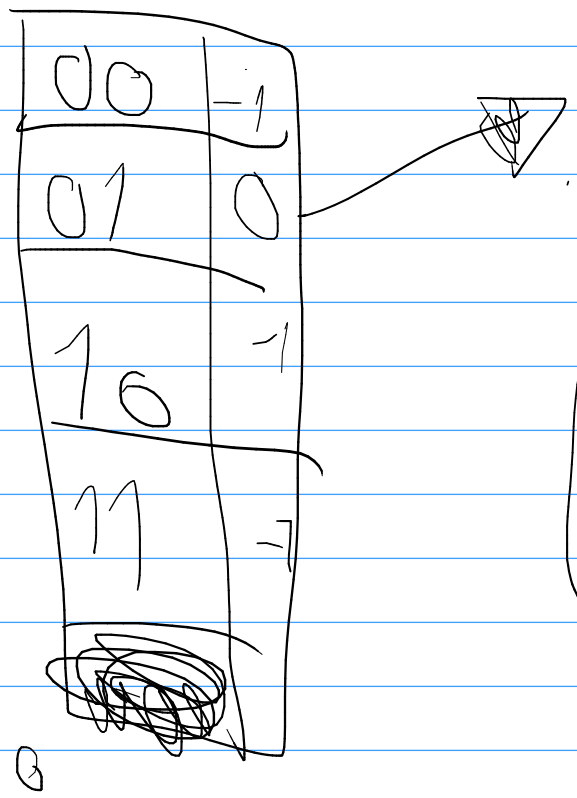


d=1 [a, b]

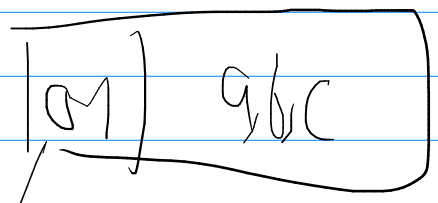
00 && 01 = 00



101



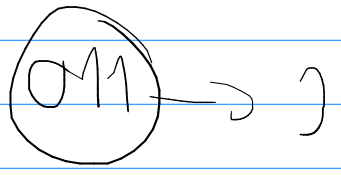
d=2



a -> 1 -> 001
b -> 5 -> 101
c -> 7 -> 001

101 001

01



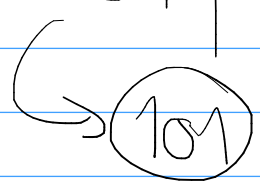
split bucket

index = 001

t = 2

pos = 0

new_index = 01 | 100



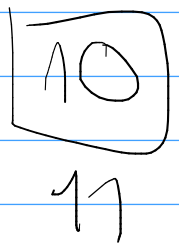
new_mask = $2^{t+1} - 1$

= 7 = 111

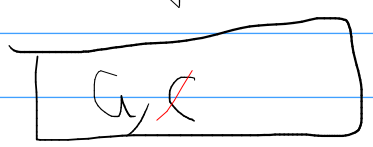
000 = 0
010 = 2
110 = 6

100 -> 4
110 -> 6
100 ->

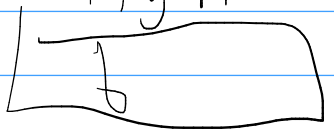
100 -> 4
000 -> 0
010 -> 2

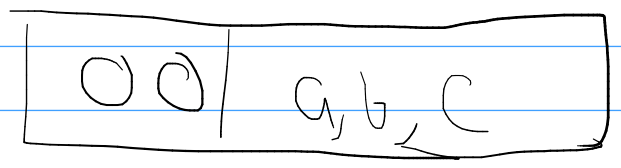


left

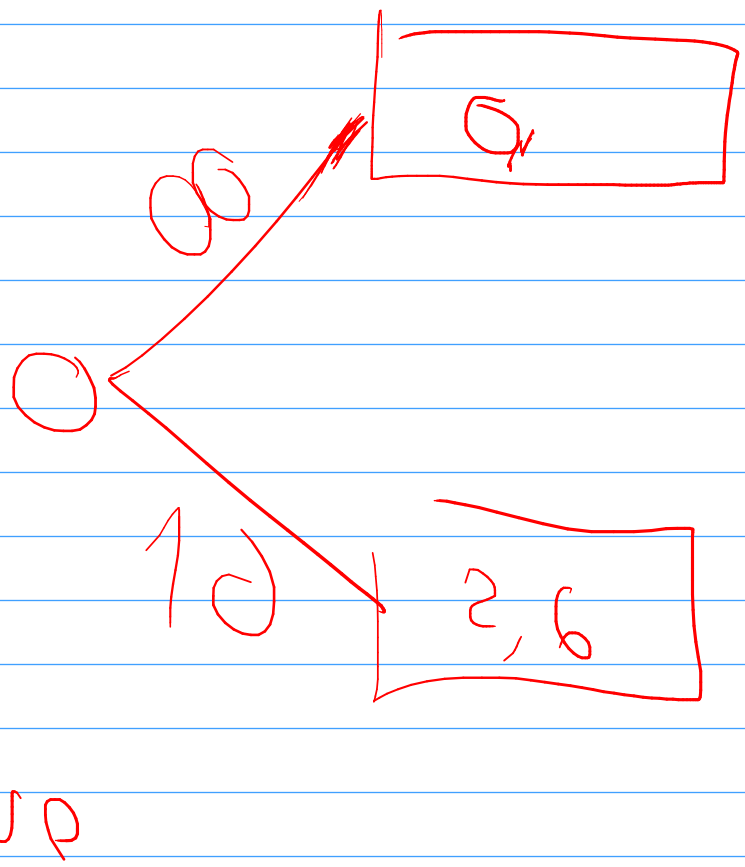


right





a -> 000
b -> 100
c ->



remap

000	1
001	
010	2
011	
100	
101	0
110	
111	

000	1
001	
010	2
011	
100	
101	0
110	-1
111	

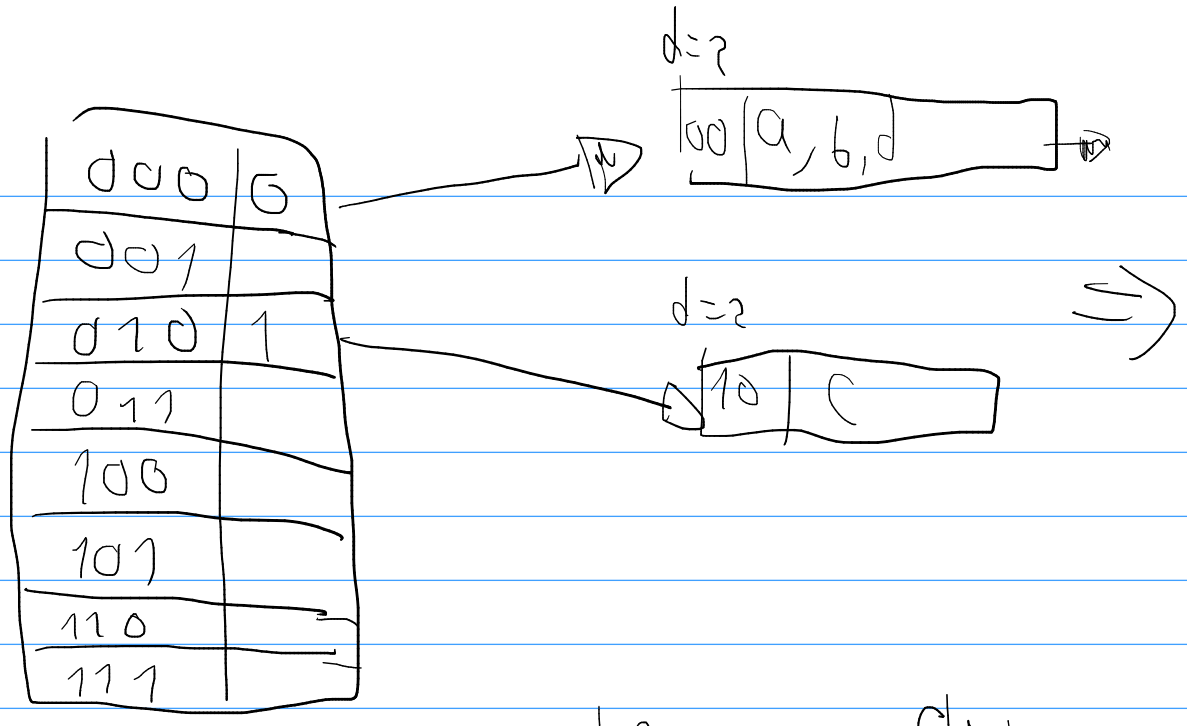
010 -> 2

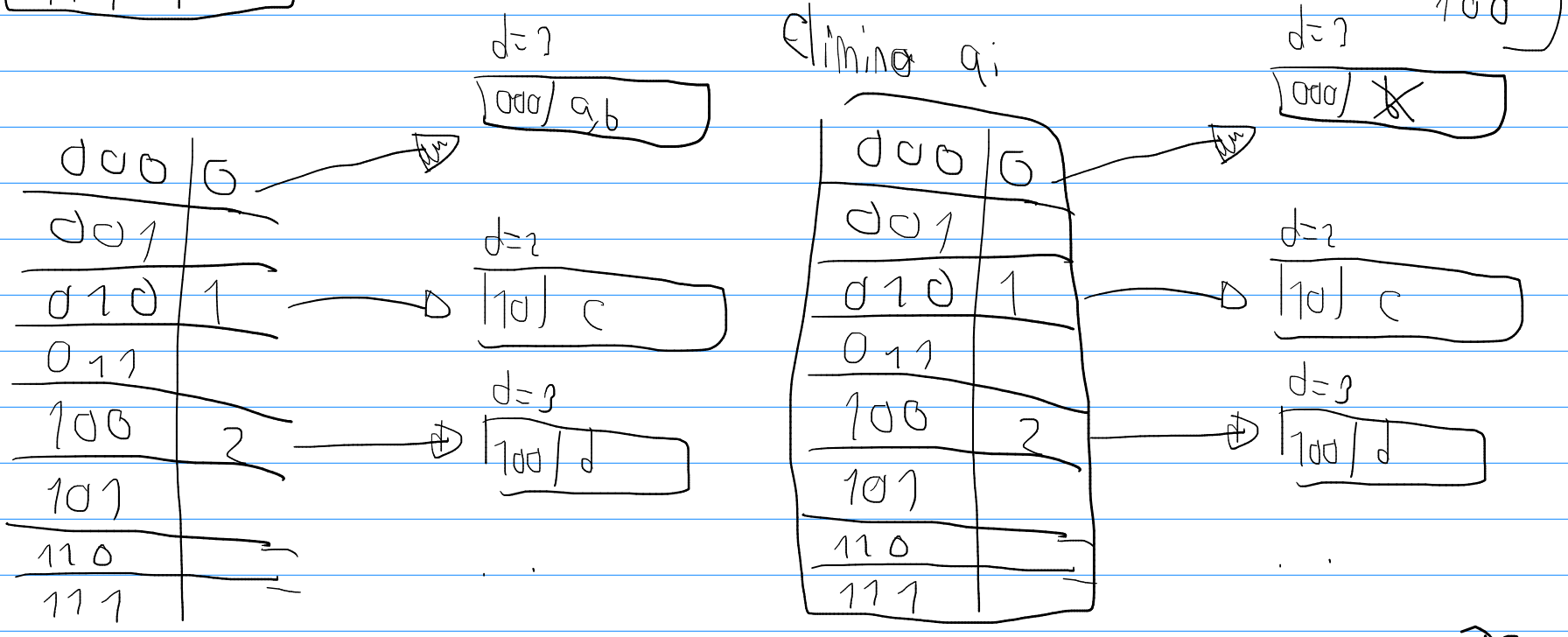
101 1

001

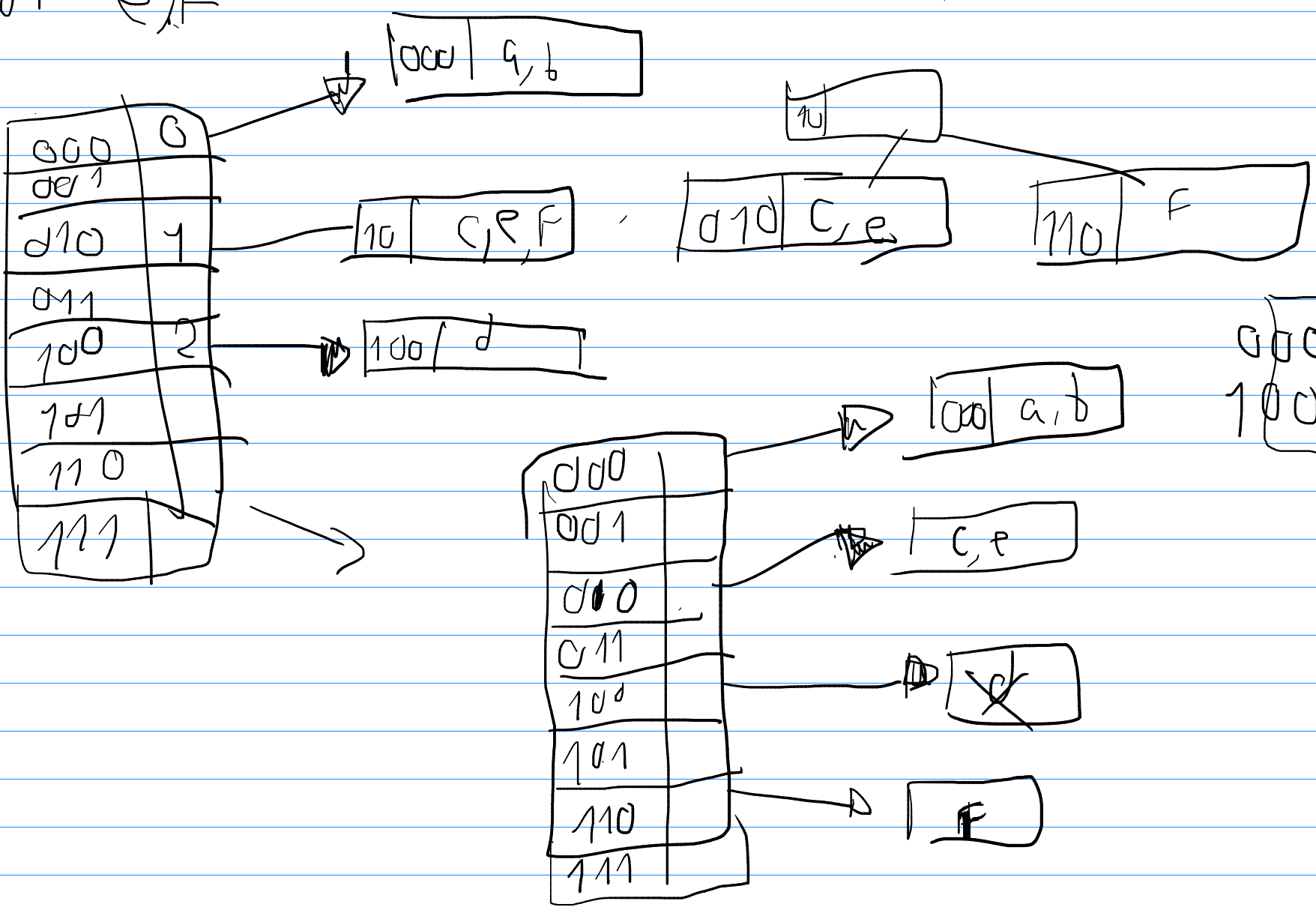
110

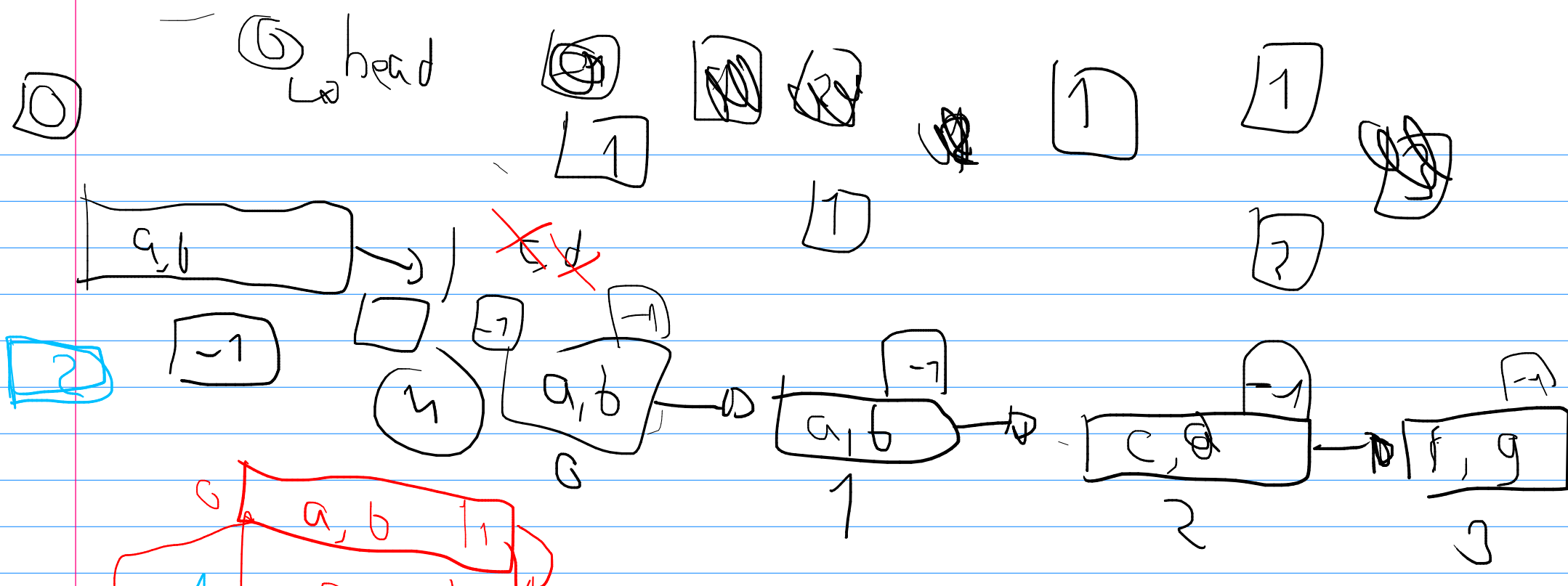
X


d=?
000 | a, b, d
d=2
10 | c
000 | 6
001
010 | 1
011
100
101
110
111


9
000 | 6
001
010 | 1
011
100 | 2
101
110
111
d=?
000 | a, b
d=2
10 | c
d=3
100 | d
Elimina a;
000 | 6
001
010 | 1
011
100 | 2
101
110
111
d=?
000 | ~~a~~
d=2
10 | c
d=3
100 | d
100

Invert e, f


000 | 6
001
010 | 1
011
100 | 2
101
110
111
000 | a, b
10 | c, e, f
100 | d
10
010 | c, e
110 | f
000 | a, b
001 | c, e
010
011 | ~~d~~
100
101
110 | f
111
000
100



Insert d

1° Inserción encadenada

1° Nos fijamos el head

Si es (-1)

↳ Creamos un nuevo bucket
la posición de ese bucket
la ponemos en el head ya que
tiene libre

Si es diferente a -1

↳ Nos ubicamos en esa posición
e insertamos, luego verificamos
si esta lleno, actualizamos el head
sino sigue siendo libre ese bucket

Para eliminar encadenado

1° Ubico el elemento

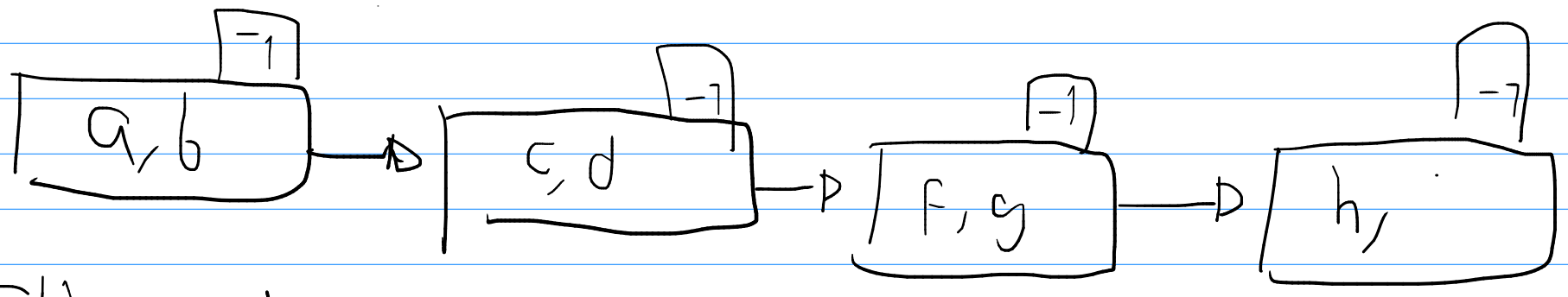
2° Guardo la posición del
registro a eliminar

3° Elimino y actualizo
el bucket sige .

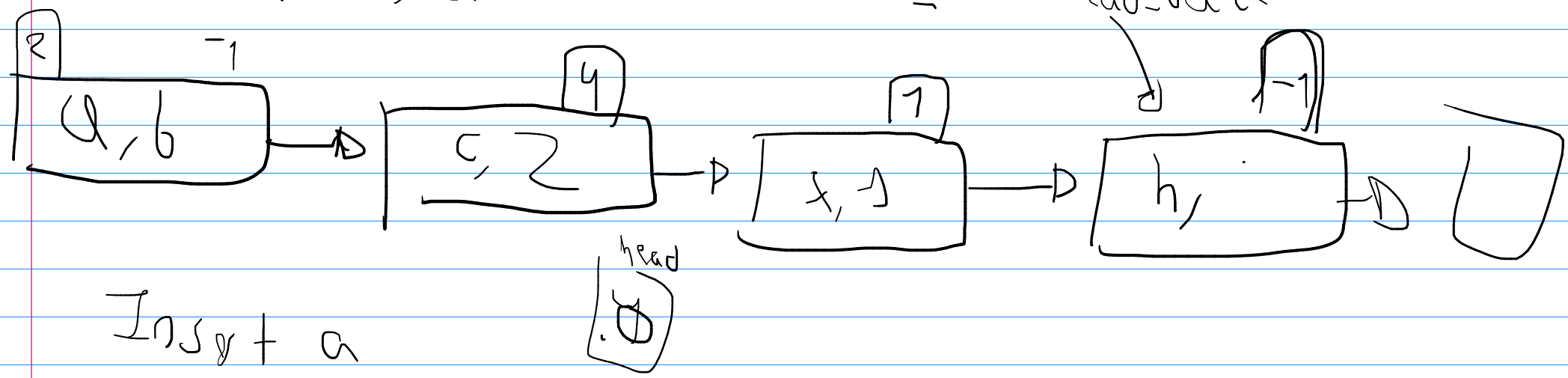
4° Me fijo si la pos es igual
al head

Si son diferentes actualizo
el head para marcar ese
bucket como libre

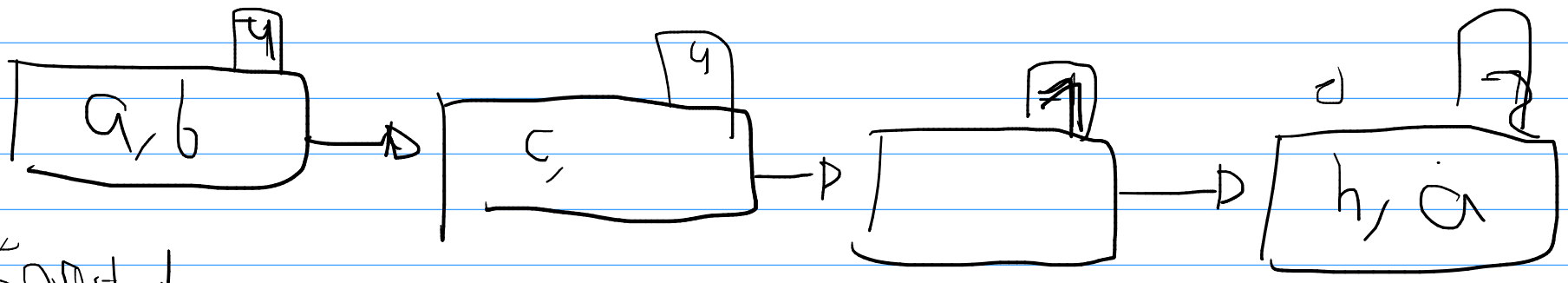
4



Elimino d, g, f, a 0 → 2 → 1 → 4



Insert a

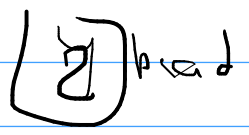


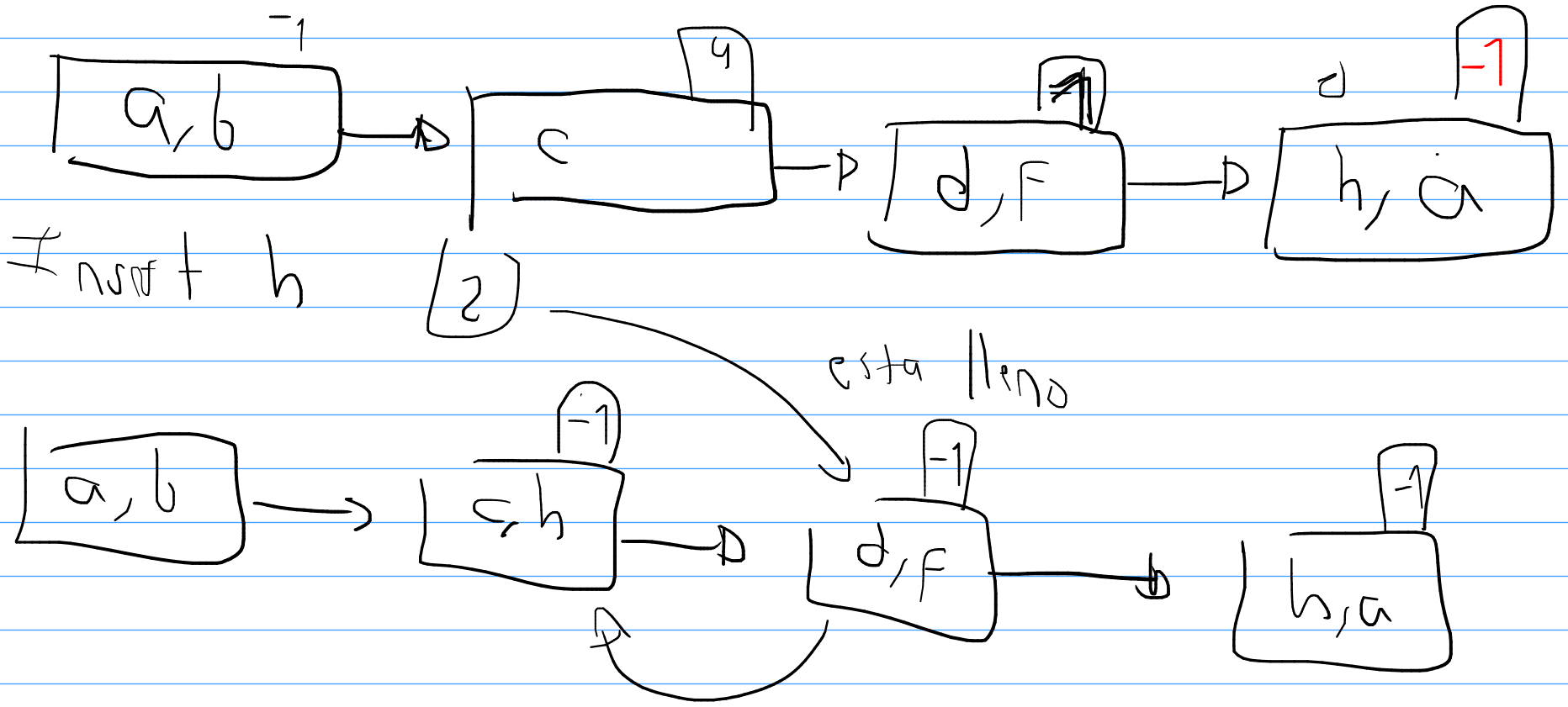
Insert f

head → 4 → esta llena

vamos al siguiente y actualizamos head

Insert d, f

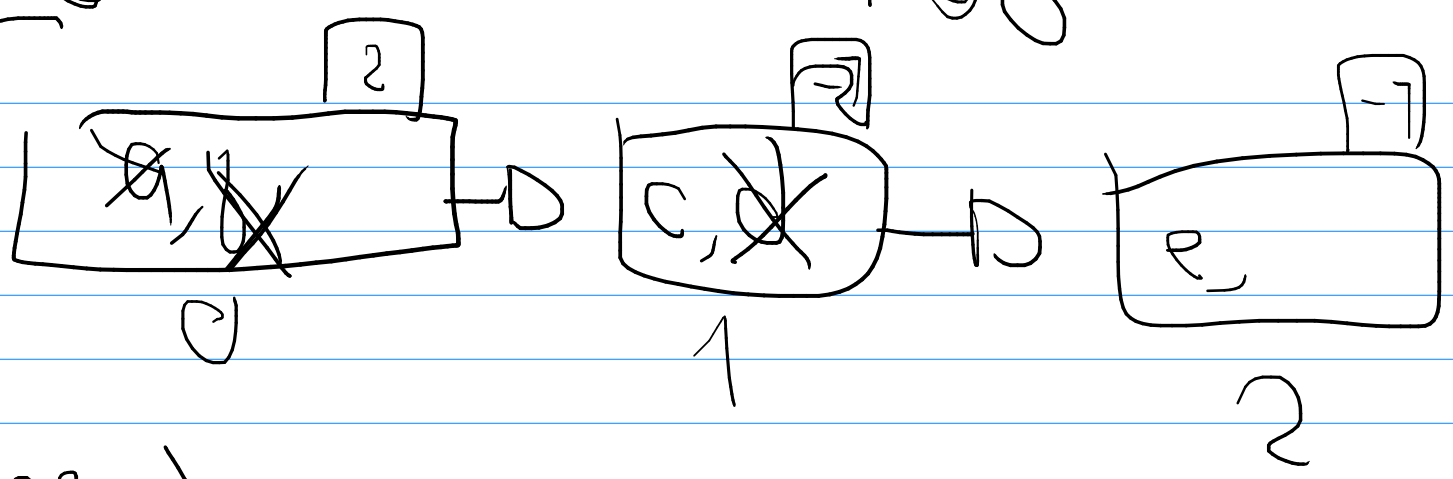




Insert h

esta llena

head = 0

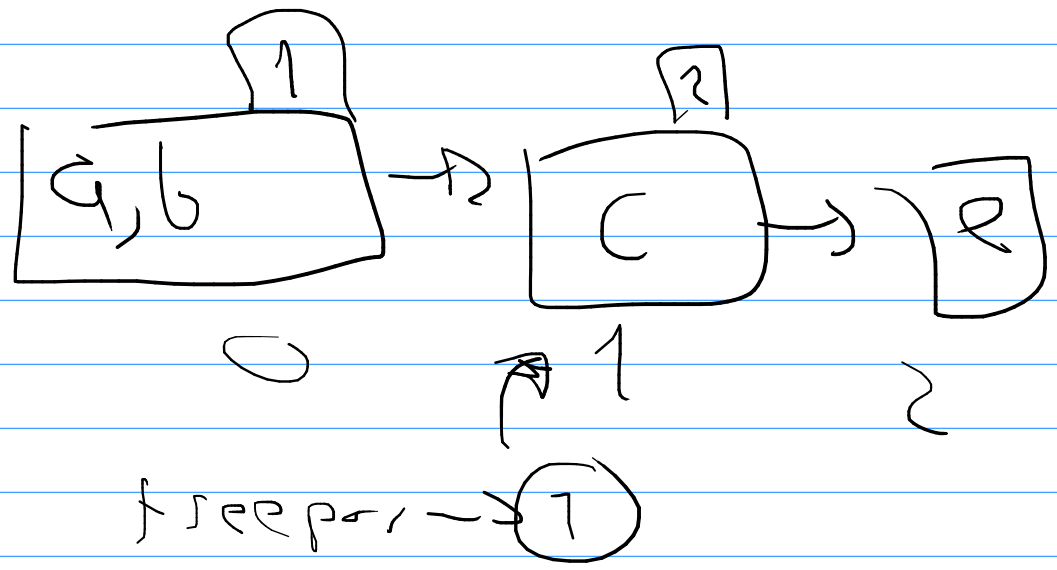


head 2

temp = 2

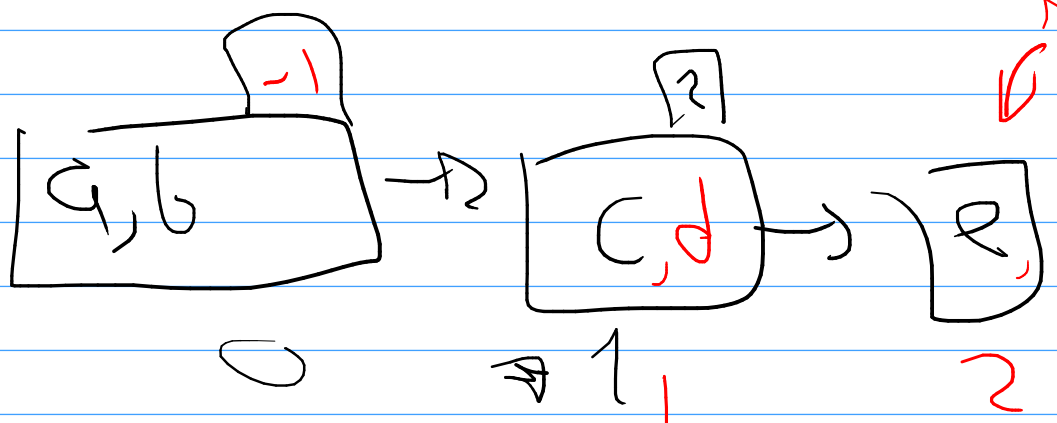
head = 1

0 -> 1 -> 2

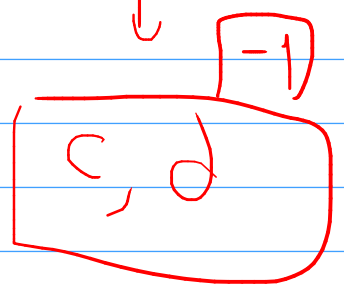


1 -> 2

head = 0

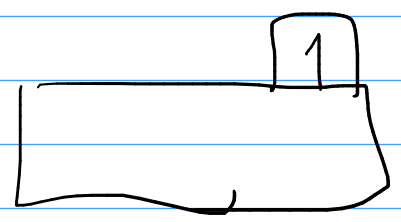


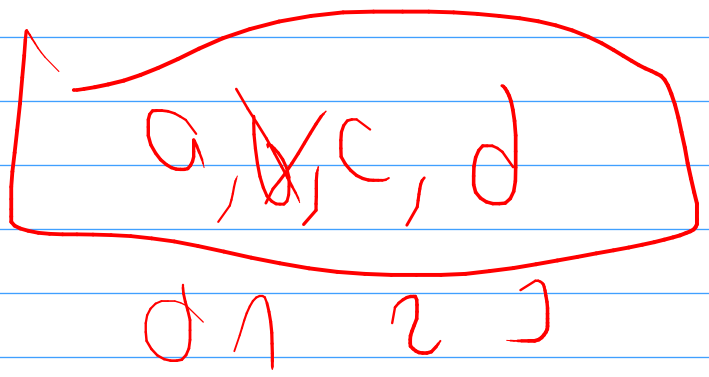
a, c, d,



2

0 -> 1

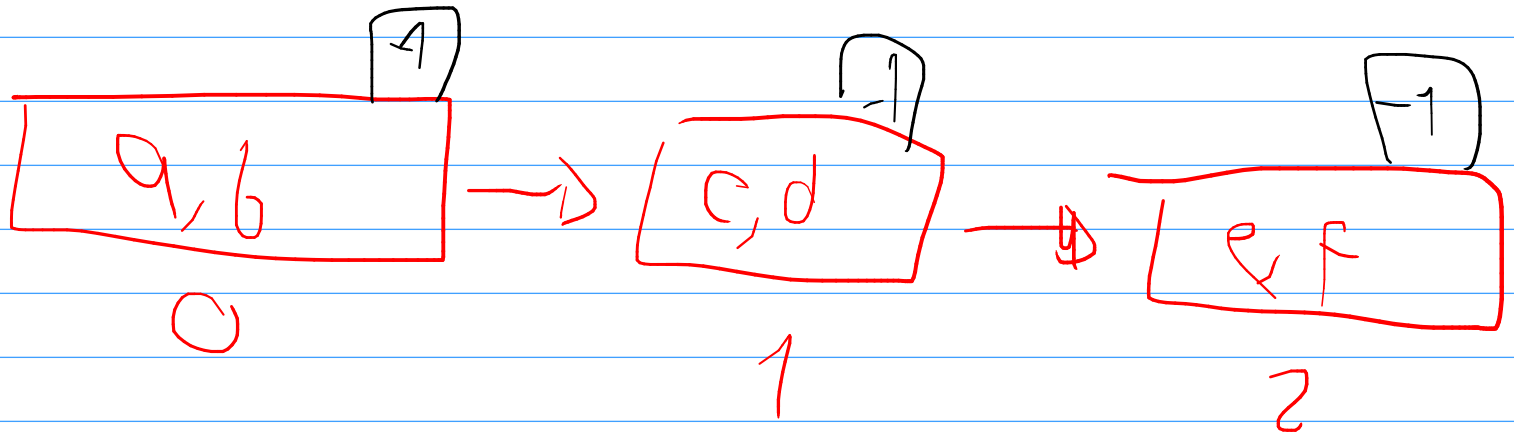


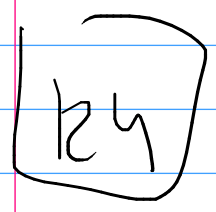


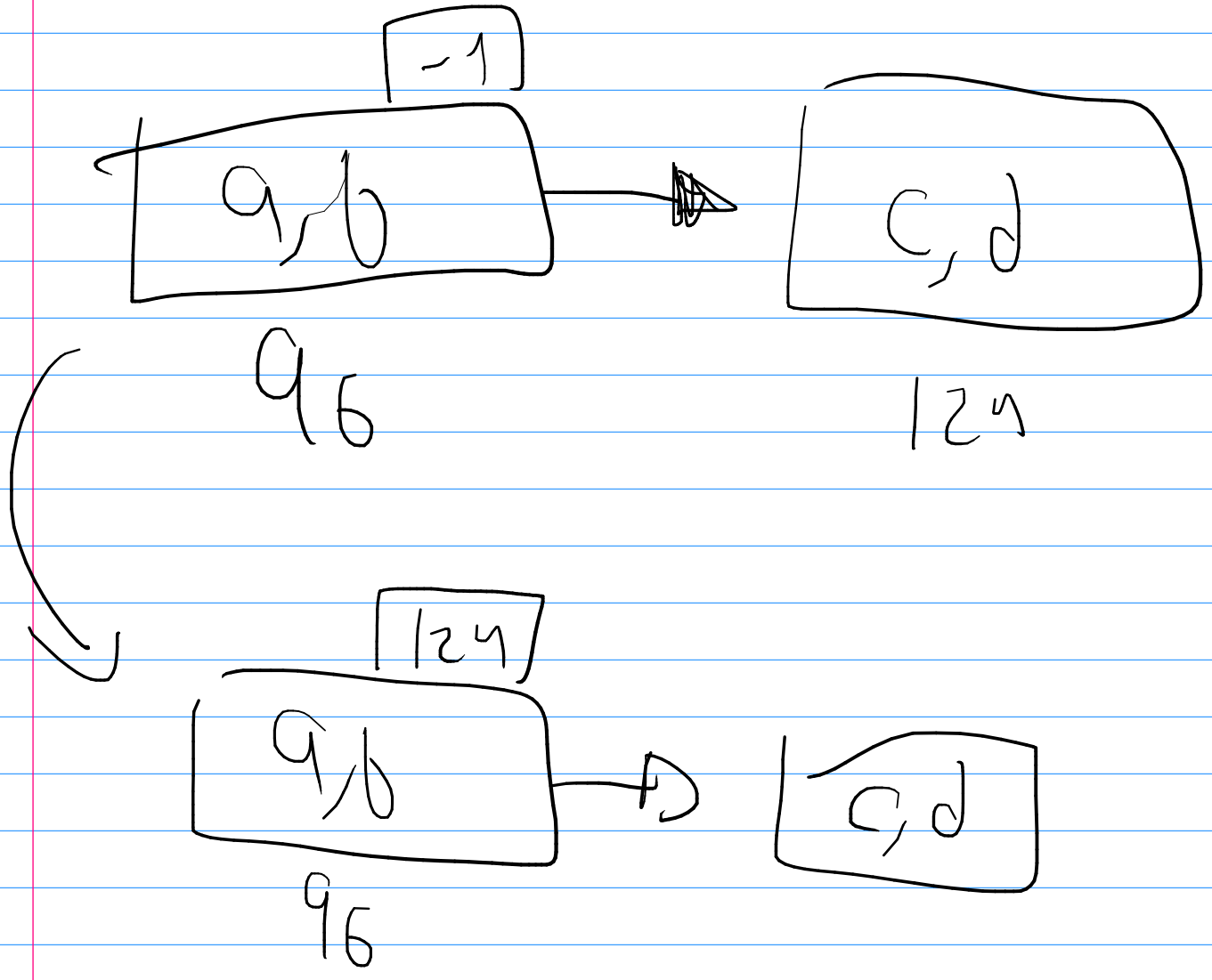
0 1 2 3

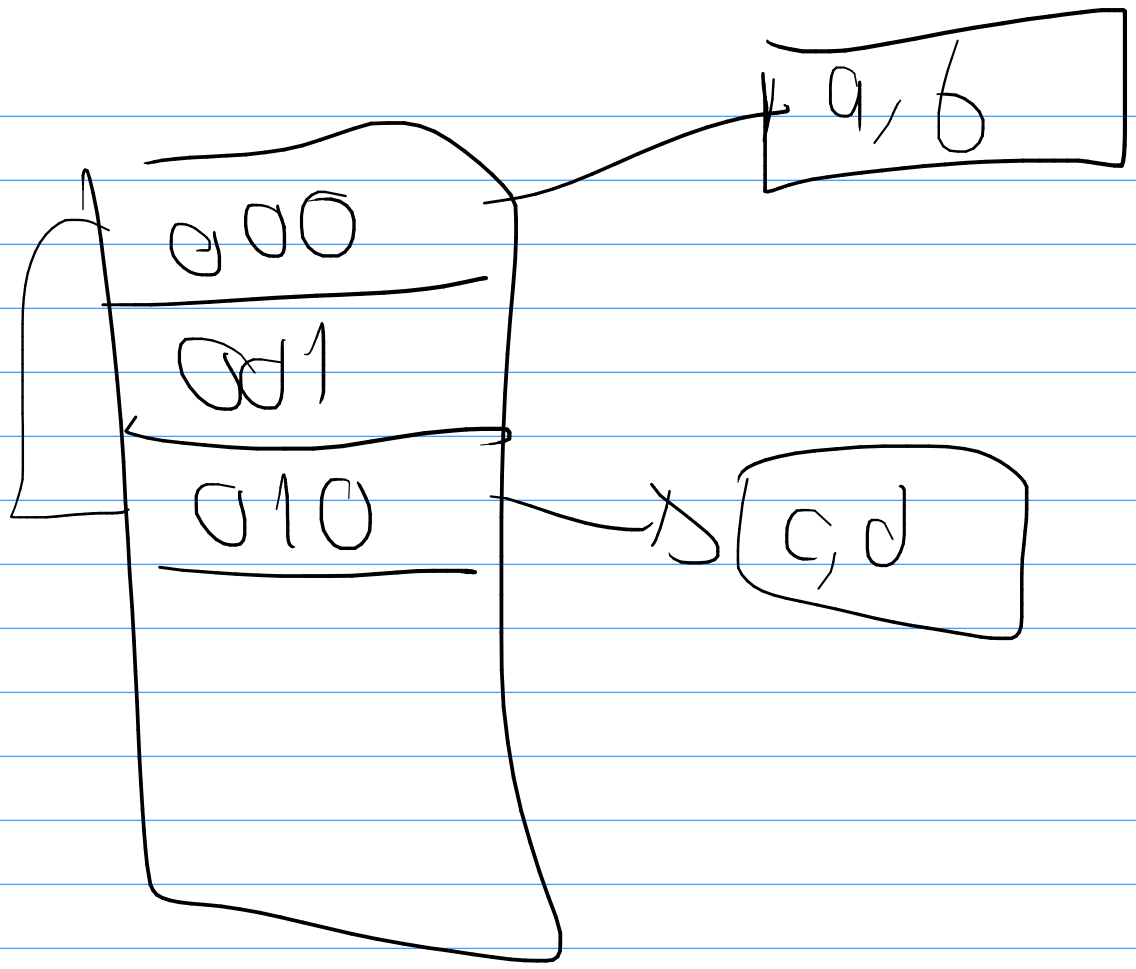
head =

brad = 2









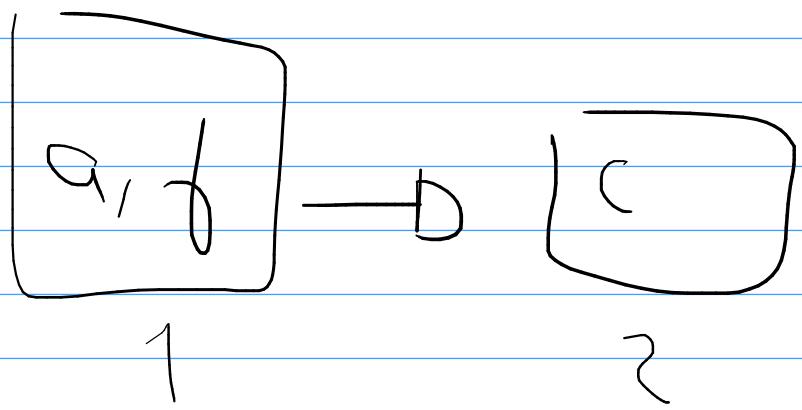
head = 2

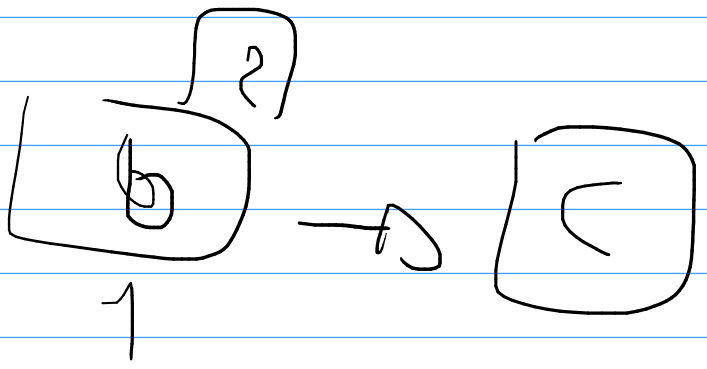
free = -1

Remover a

head = 1

free = 2






```
 --------- Bucket 2
head: 96
size: 2
free pos: -1
2 2  -> 2
profundidad local: 3
```

```
head: 36
size: 1
free pos: 96
2  -> 2
profundidad local: 3
```