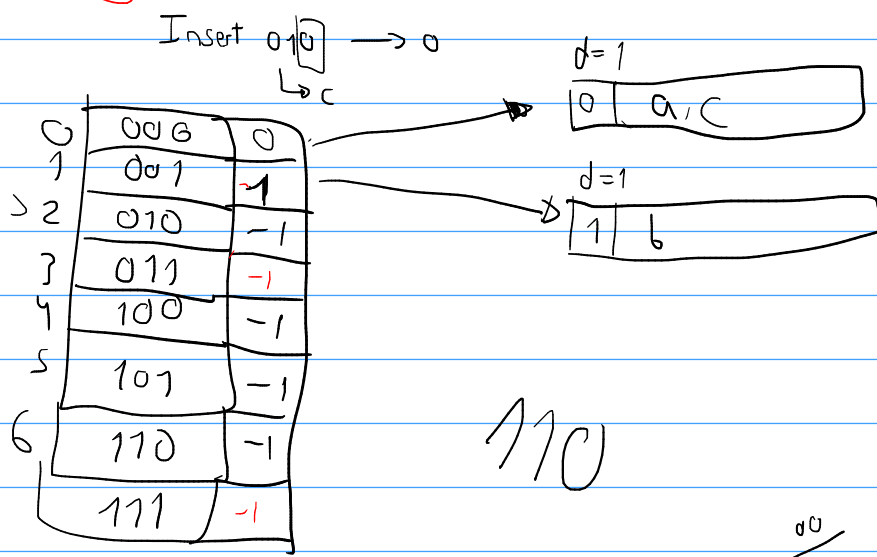
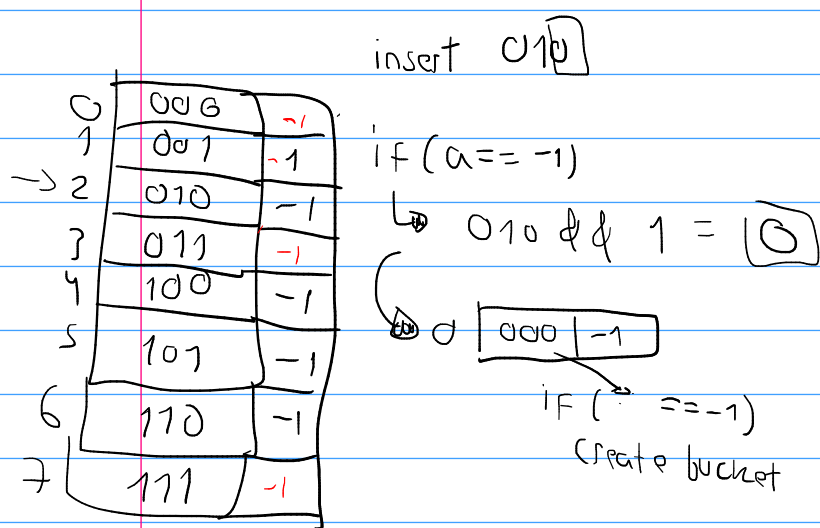
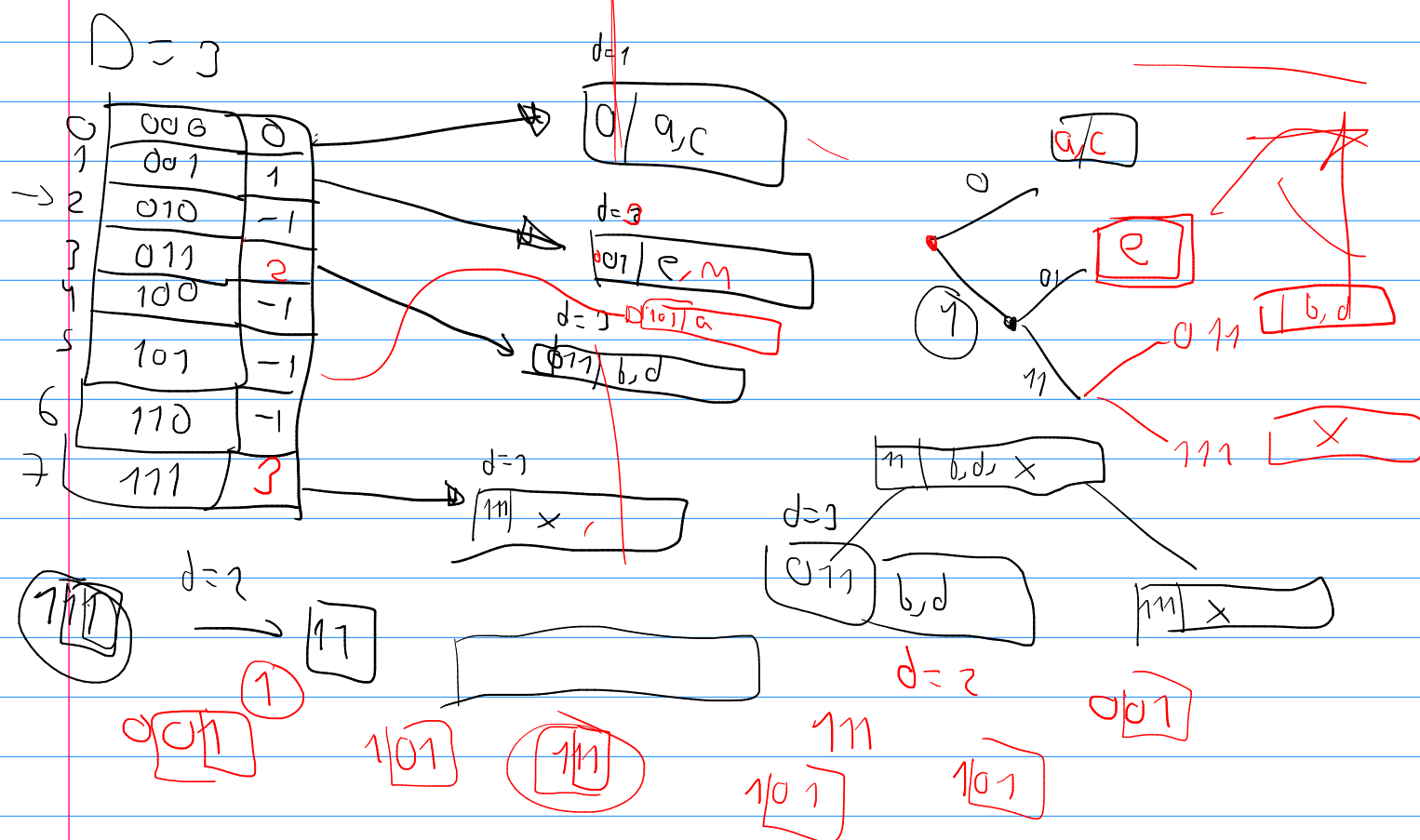


010 → a 000 → x
 011 → b 111 → d 101 → e

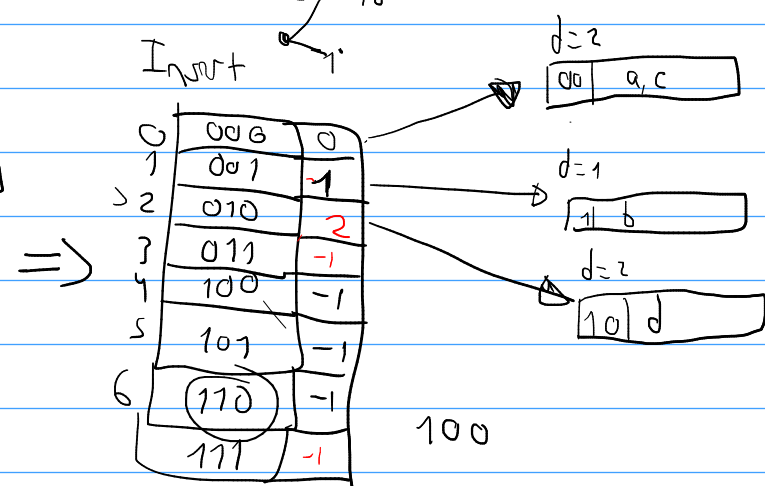
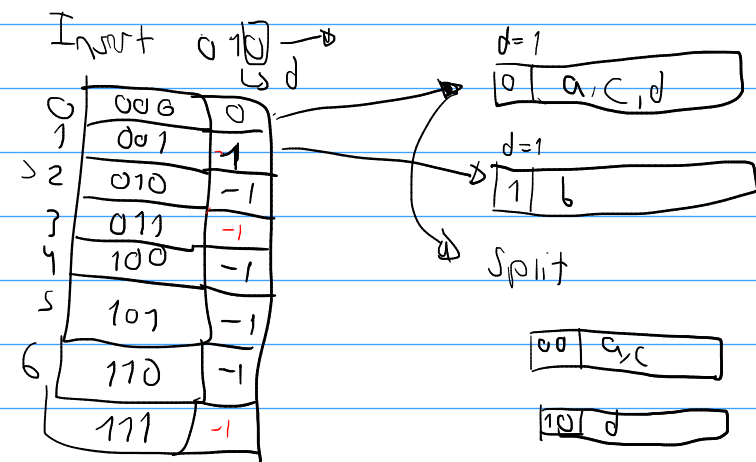


40% 8

010

010

mask 2 = 1



000

110

Insert 110 → d=2 ⇒ 10

d=2

mask 2 = 11

110 & 11 = 10

< t=2

mask 2 = 11

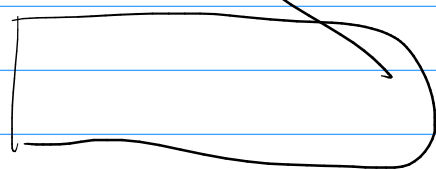
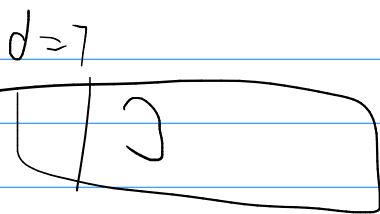
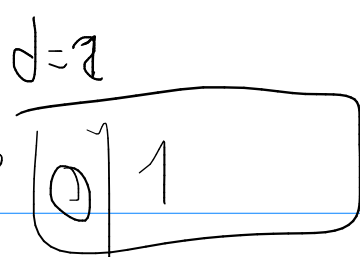
t=1

2

$D=4$

00	29
01	0
10	-7
11	-7

0



10

0 10

For (Buckets)

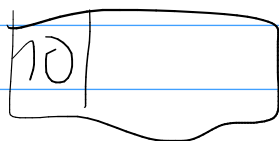
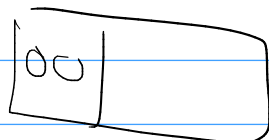
a b c
00 10 10

Split (bucket, ^{mark}~~index~~, t)

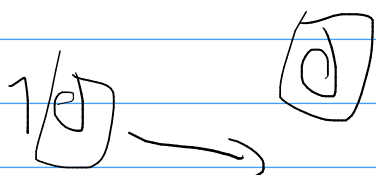
t=2
11

bucket

a, b, c



mark = 0



d=1
t=1

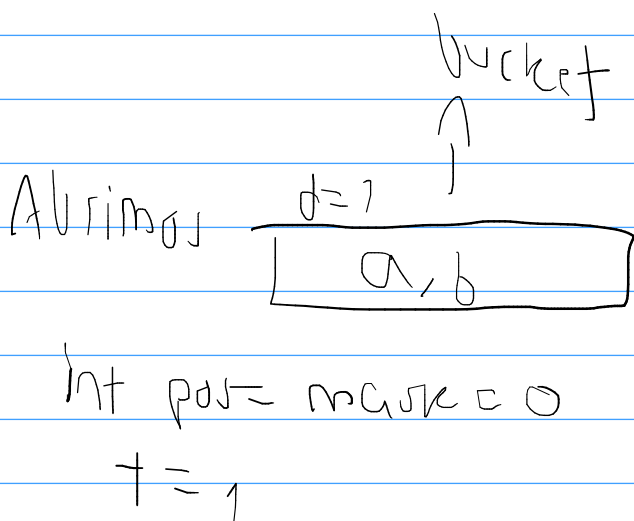
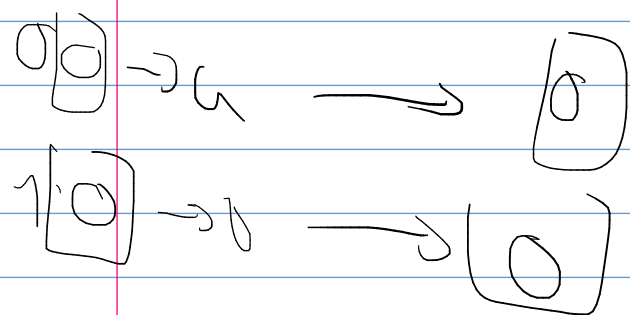
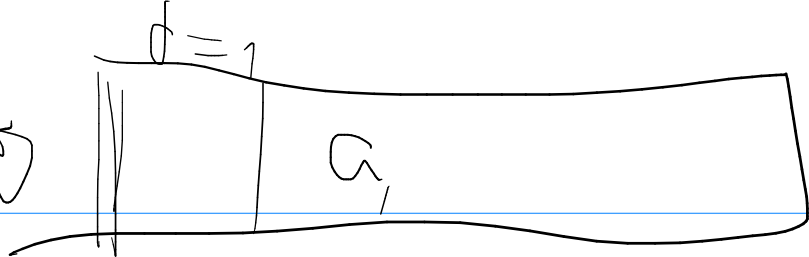
00
10

00

t=1

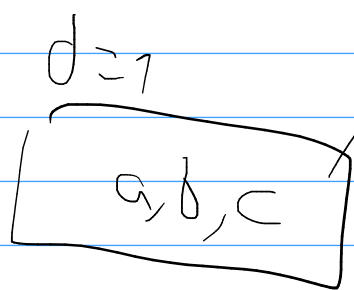
mark

00	0
01	-1
10	-1
11	-1



key = b
Index = b
mark = 0

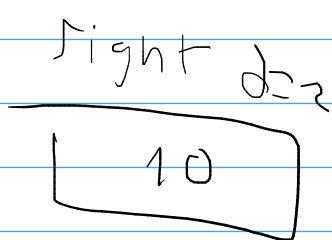
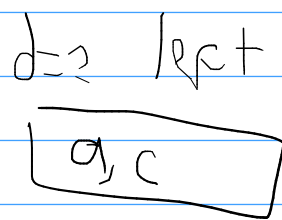
00	0
01	-1
10	-1
11	-1



mark = 0
pos = 0
t = 1

len = 0
new_index = 0 | 10
= 10

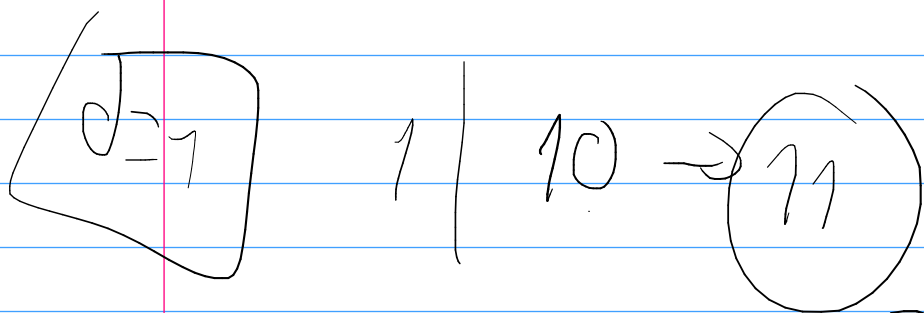
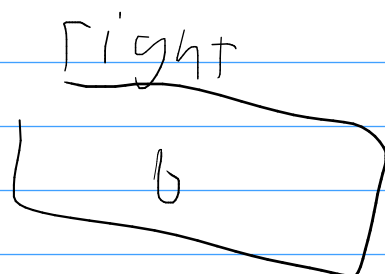
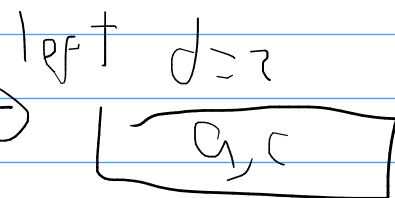
New_mask = 1 | 10 = 11



value
a → 00 → 00
b → 10 → 10
c → 00 → 00

⇒

00	0
01	-1
10	-1
11	-1



000

d = 2

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

$d=2$

a, c

Insert 100

d^c

$d=2$

10

index = 100
mask = 0
t = 1
d = 2
pos = 0
mask = 1

while ()
1st Iteration
mask = 11
mask = 100 & 11
mask = 00
pos = 0
d = 2
t = 2
scale del bump

d
 000
 $a = 110$
 $b = 010$
 $c = 010$

a, b, c

a, c, d

Split

new_index = 00 | 100 = 100

new_mask = 111

$010 \rightarrow$
 010
 010

$d=2$

$010 / 100 = 110$
2

new_index = 010 | 100 = 110

left $d=3$

a, c

Right $d=3$

d

left $d=3$

a, c

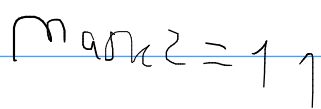
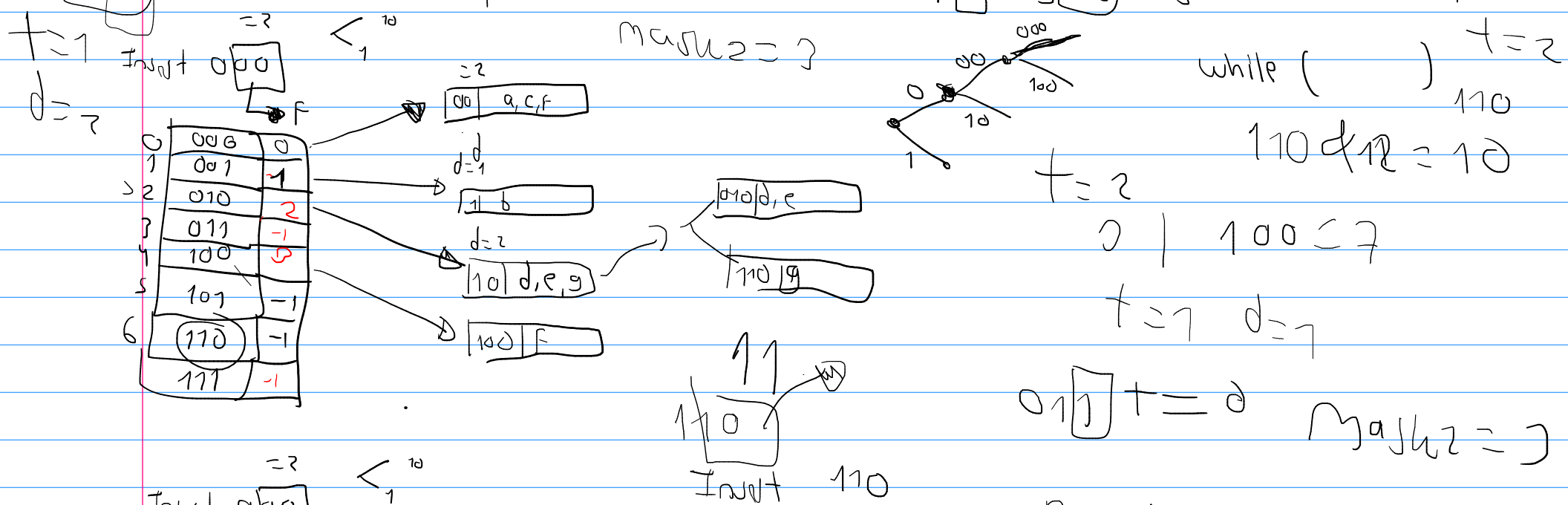
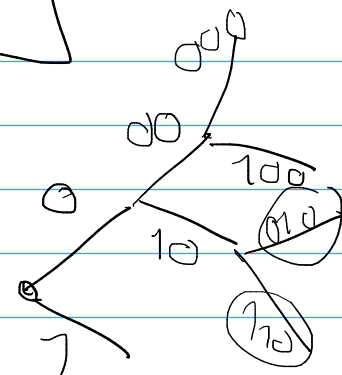
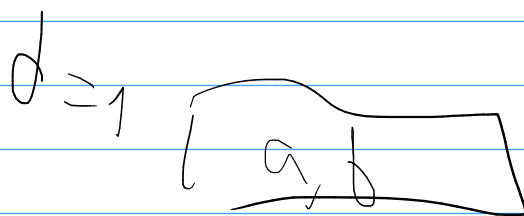
000	0
001	-1
010	1
011	-1
100	2
101	-1
110	-1
111	-1

$d=2$

b

$d=3$

d


$$+ = = d$$


000	
001	
010	
011	
100	
101	
110	
111	

101

$d=2$

$a \rightarrow 1 \rightarrow 001$

$b \rightarrow 5 \rightarrow 101$

$c \rightarrow 7 \rightarrow 001$

00	-1
01	0
10	-1
11	-1
100	

101 | a, b, c

101

001

01

(011) \rightarrow)

split

bucket

index = 001

$t=2$

pos = 0

new_index = 01 | 100

\rightarrow (101)

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

= 7 = (111)

left

a, b, c

right

b

10

11

00 | a, b, c

$a \rightarrow 000$

$b \rightarrow 100$

$c \rightarrow$

000 = 0

010 = 2

110 = 1

100 \rightarrow 4

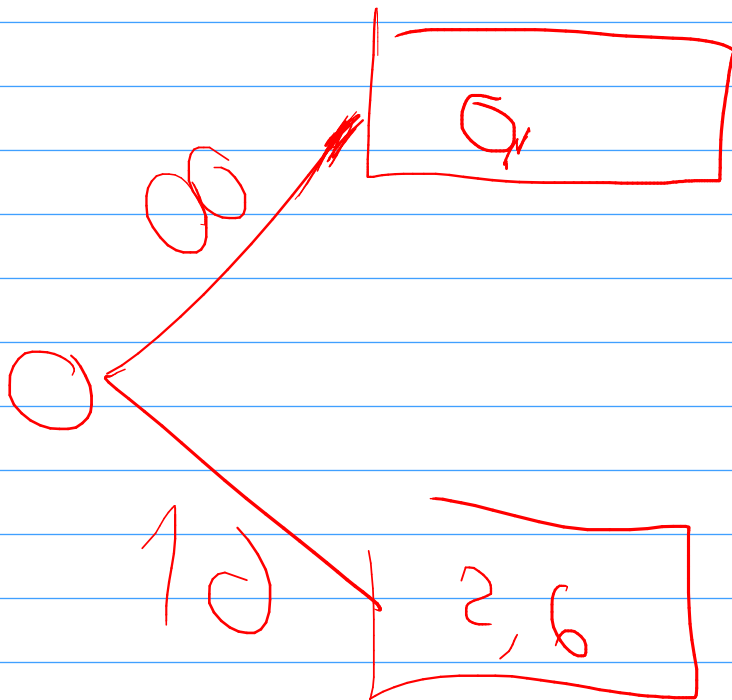
110 \rightarrow 6

100 \rightarrow

100 \rightarrow 4

000 \rightarrow 0

010 \rightarrow 2



remains

000	1
001	
010	2
011	
100	
101	0
110	
111	

0	0, 2, 0
--------------	---------

00	0, 0
----	------

00	2, 6
---------------	------

1	5, 1, 0
--------------	---------

00	1	3
00	1	1

01	5, 1
----	------

010 - 2

101 1

001

110

X

010 2

011

100

101

110

111

0

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

0

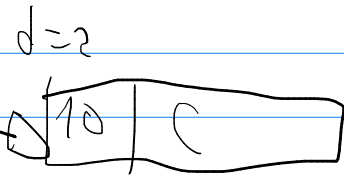
2, 6

0

00
0
10

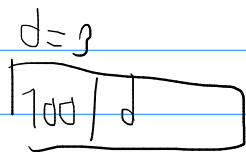
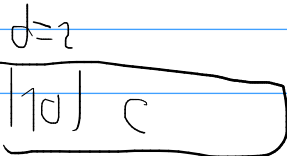
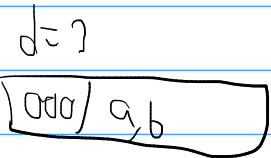
00
0
10

000	0
001	
010	1
011	
100	
101	
110	
111	



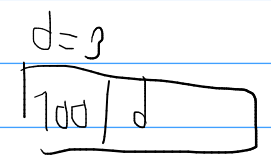
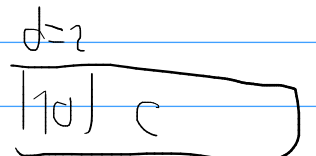
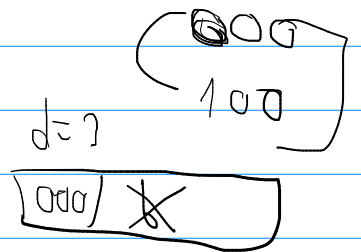
q

000	0
001	
010	1
011	
100	2
101	
110	
111	



eliminating q_i

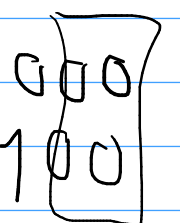
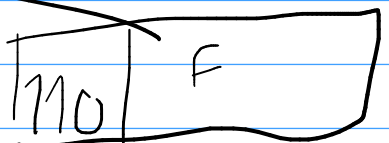
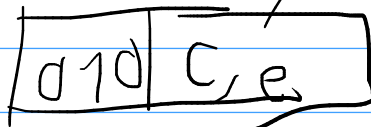
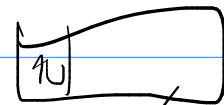
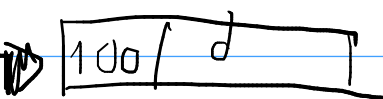
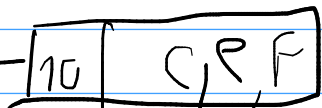
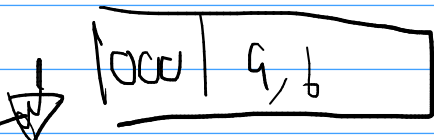
000	0
001	
010	1
011	
100	2
101	
110	
111	



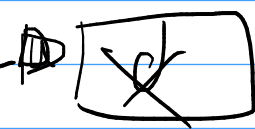
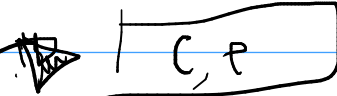
Insert e, f

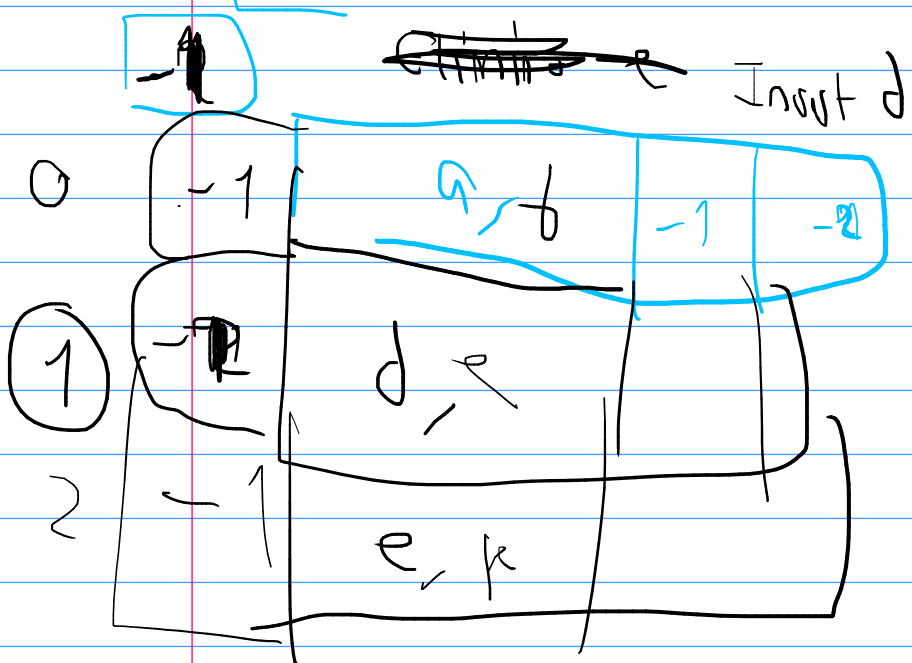
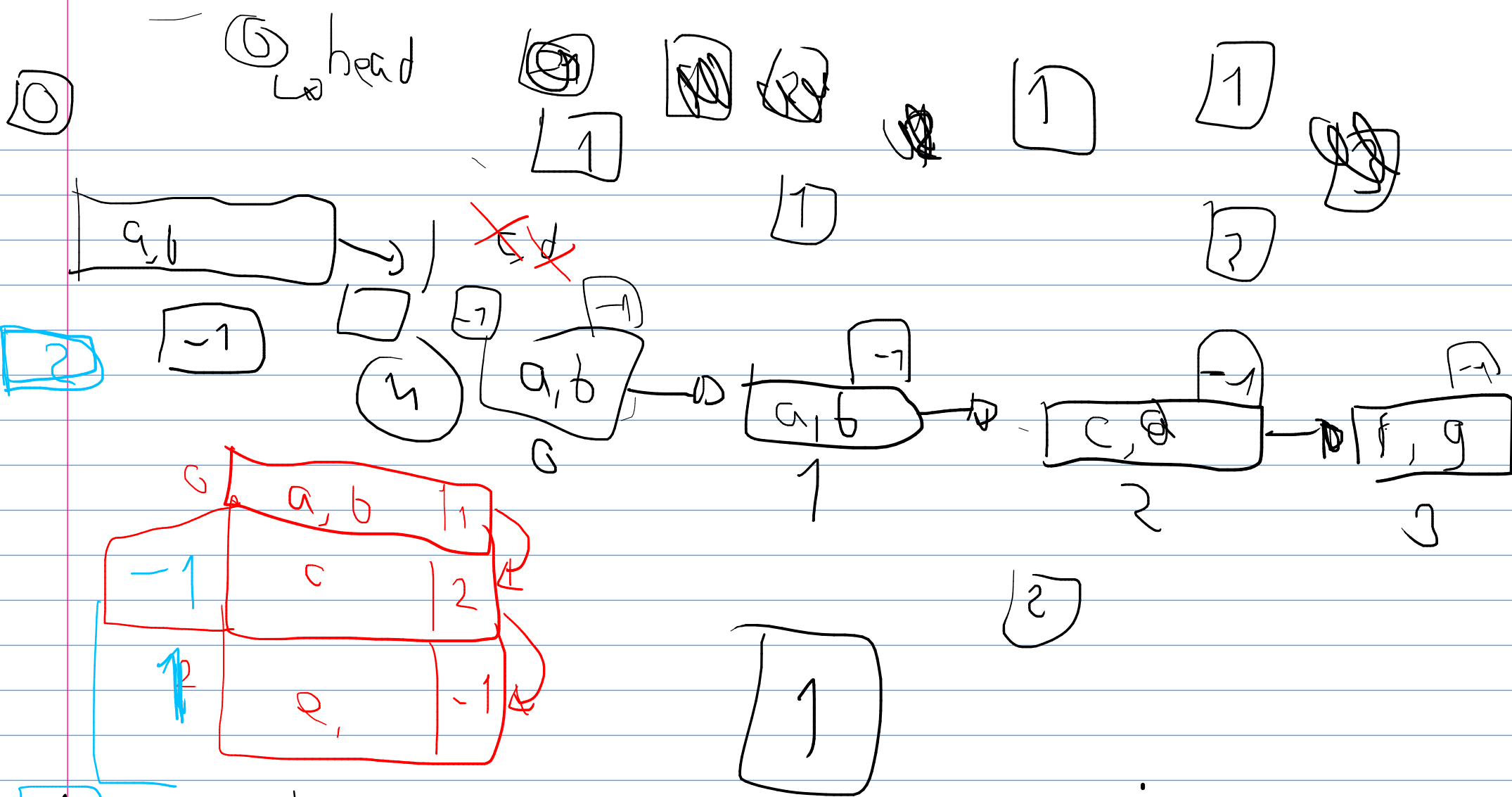
$\uparrow e_{hp}$

000	0
001	
010	1
011	
100	2
101	
110	
111	



000	
001	
010	
011	
100	
101	
110	
111	





Para eliminar en cadena

1° Ubico el elemento

2° Guardo la posición del registro a eliminar

3° Elimino y actualizo el bucket si se

4° Me fijo si la pos es igual al head

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

Insertión en cadena

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`

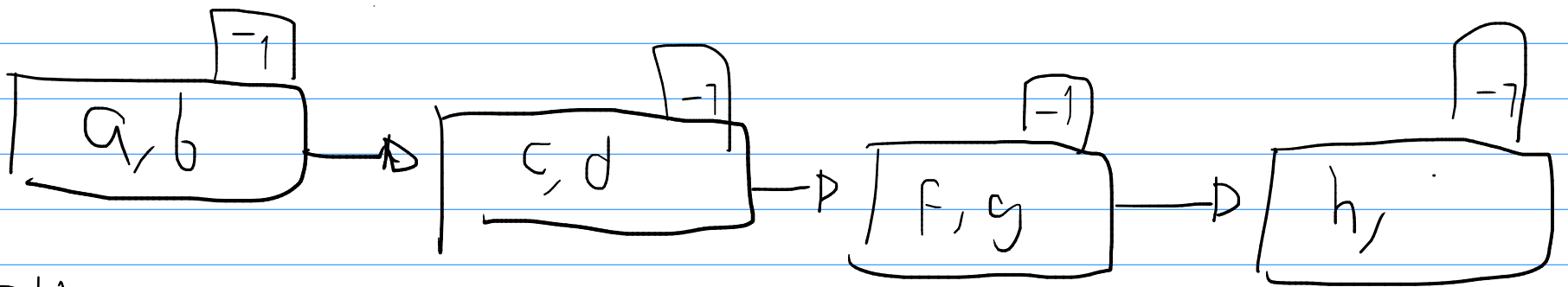
↳ Nos ubicamos en esa posición

e insertamos, luego verificamos

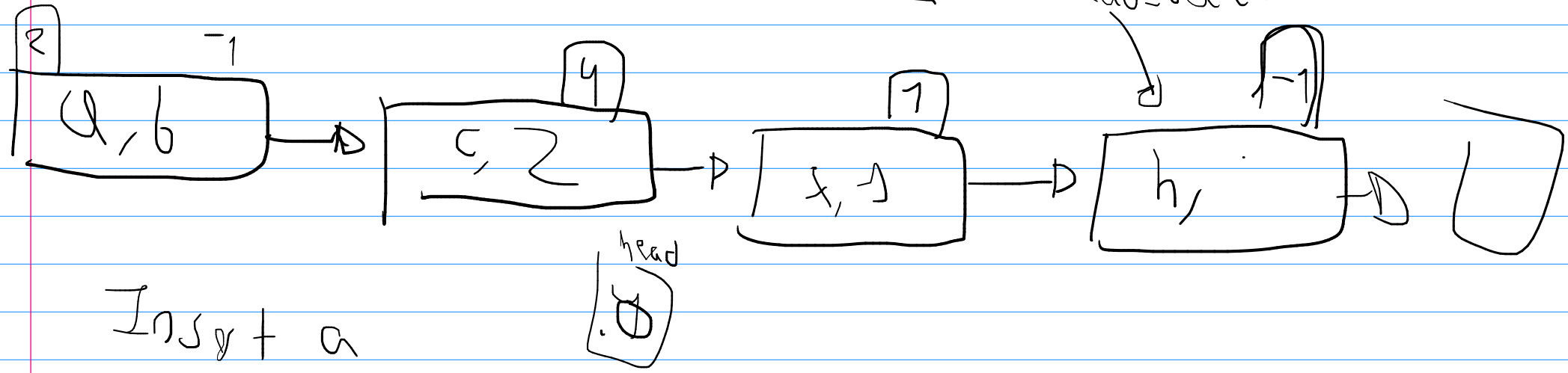
si esta lleno, actualizamos el head

si no sigue siendo libre ese bucket

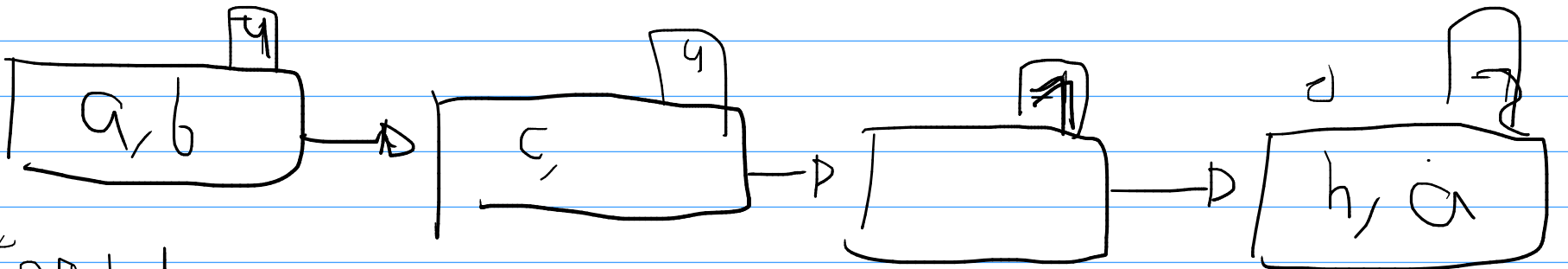
4



Elimino d, g, f, c $0 \rightarrow 2 \rightarrow 1 \rightarrow 4$



Insert a

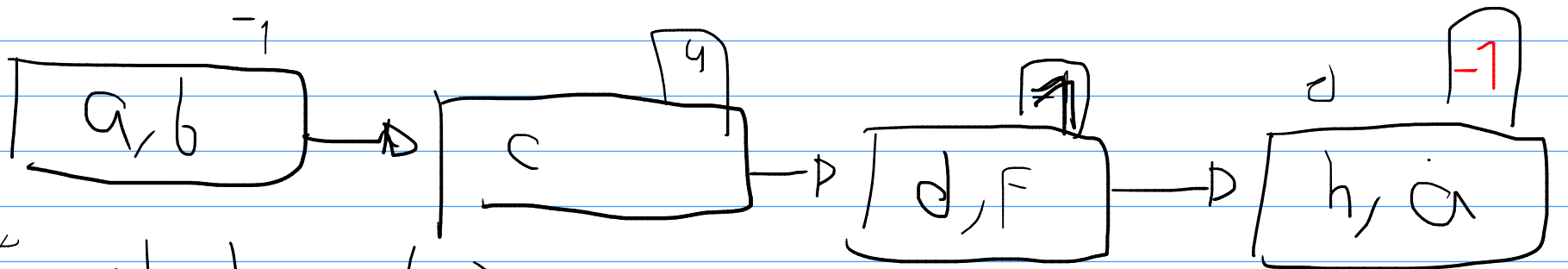


Insert f

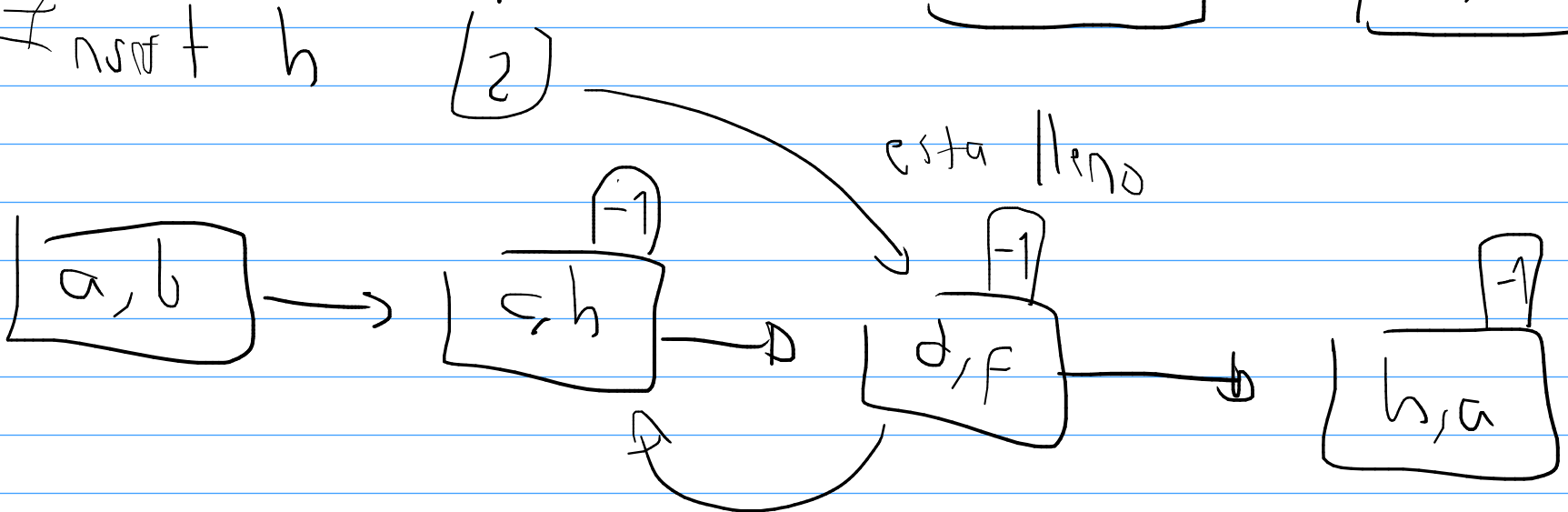
head $\rightarrow 4 \rightarrow$ esta lleno

vamos al siguiente y actualizamos head

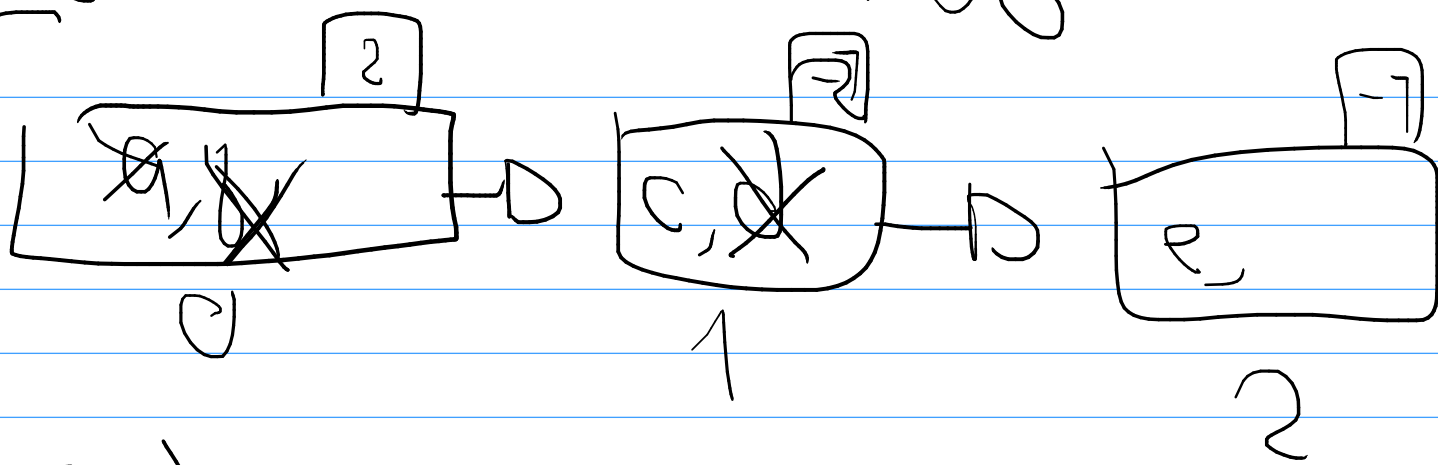
Insert ~~d~~, f $[2]$ head



Insert h



head = 0

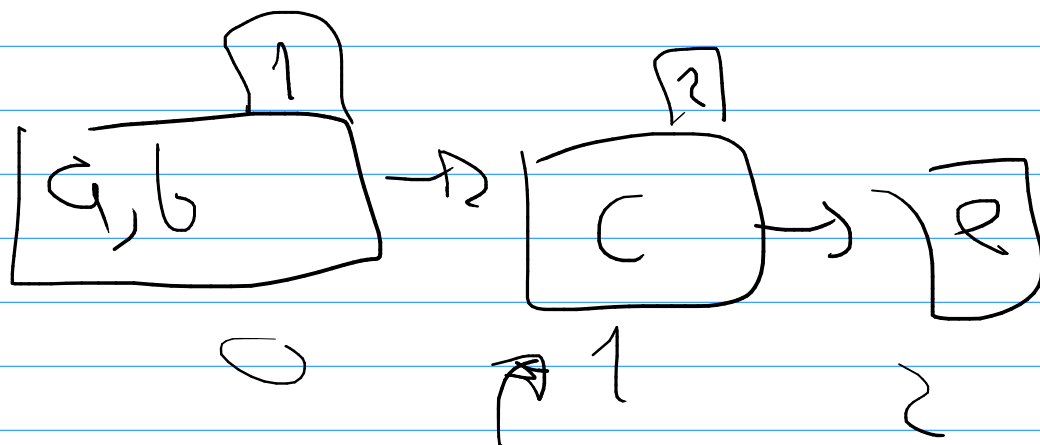


head = 2

0 -> 1 -> 2

temp = 2

head = 1

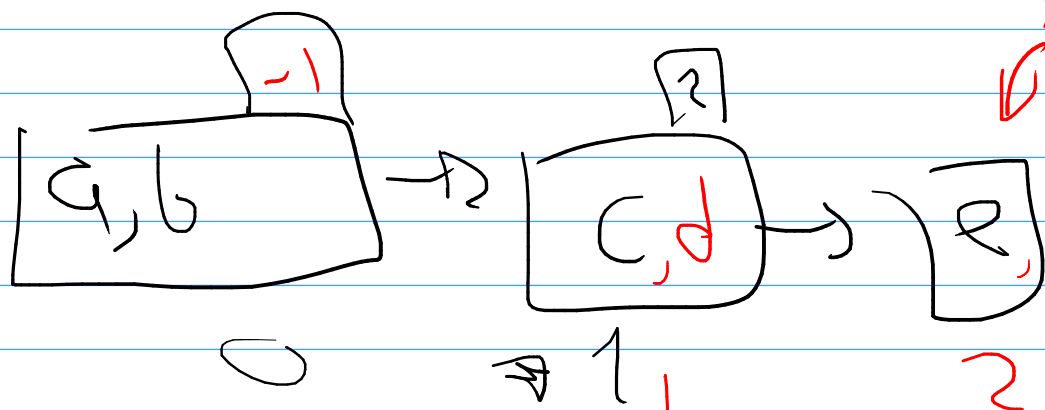


0 -> 1 -> 2

freeptr -> 1

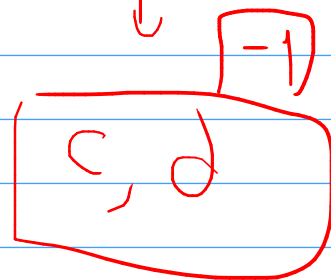
1 -> 2

head = 0



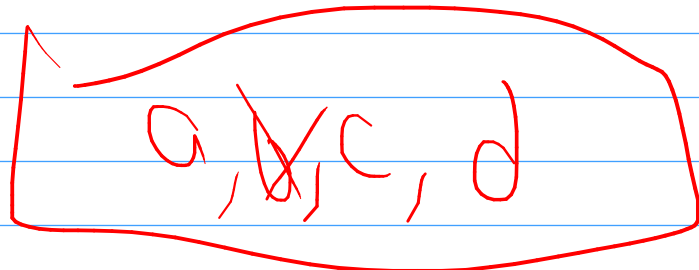
Free bucket

a, c, d,

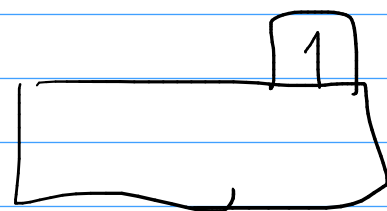


2

0 -> 1

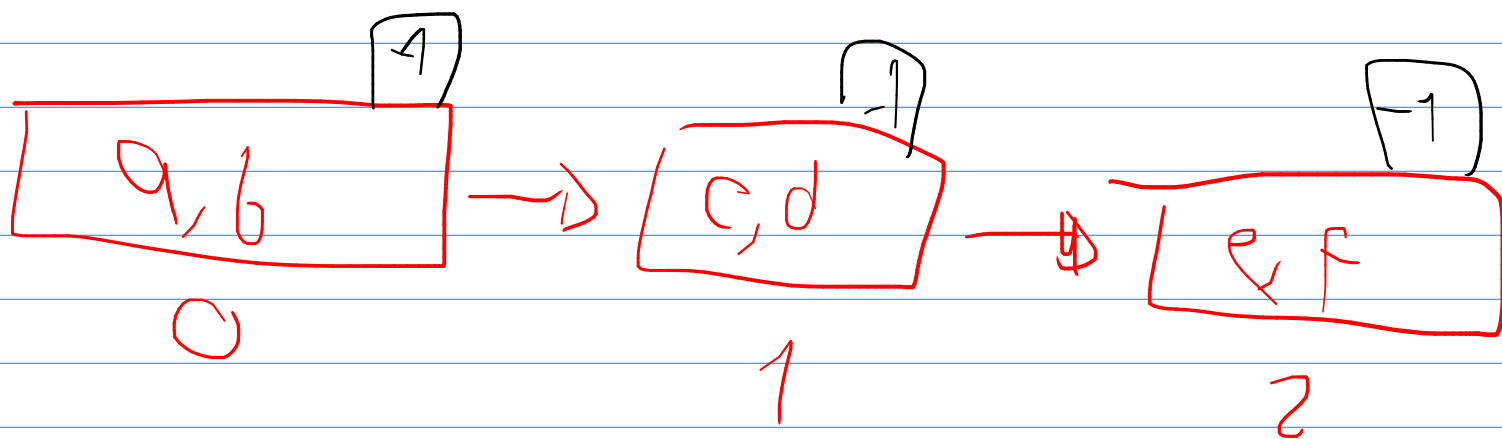


0 1 2

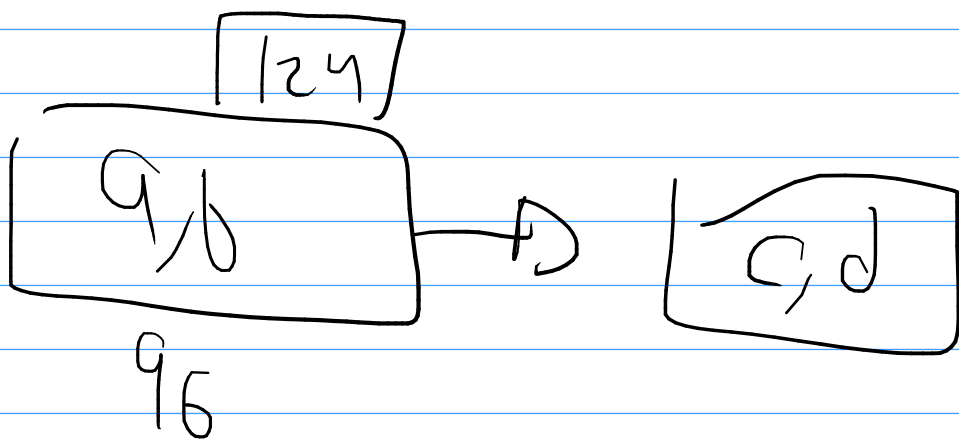
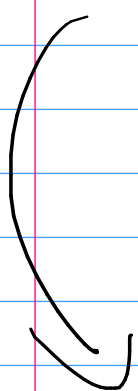
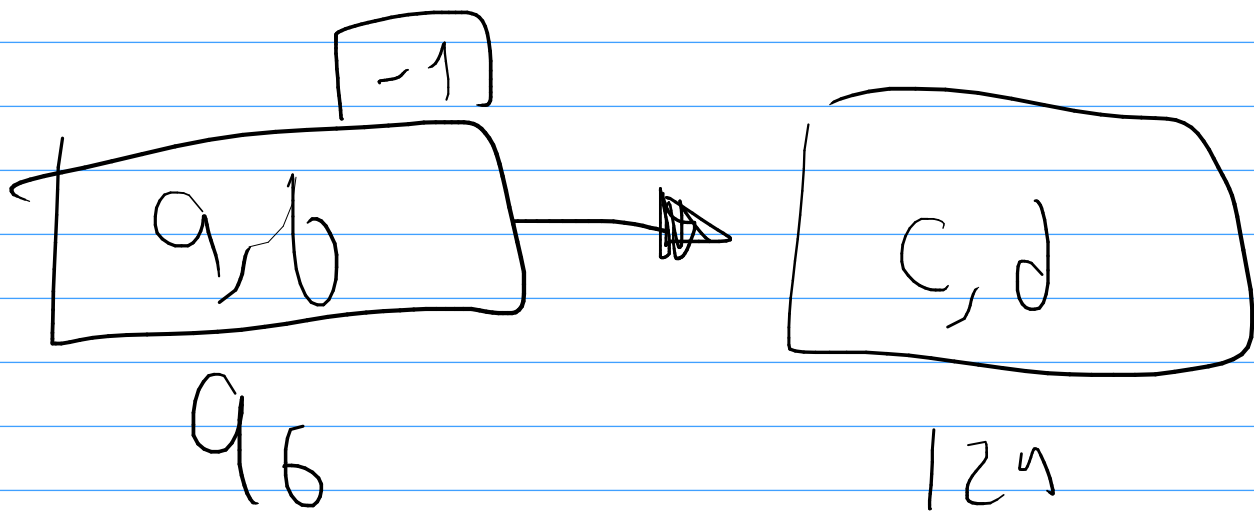


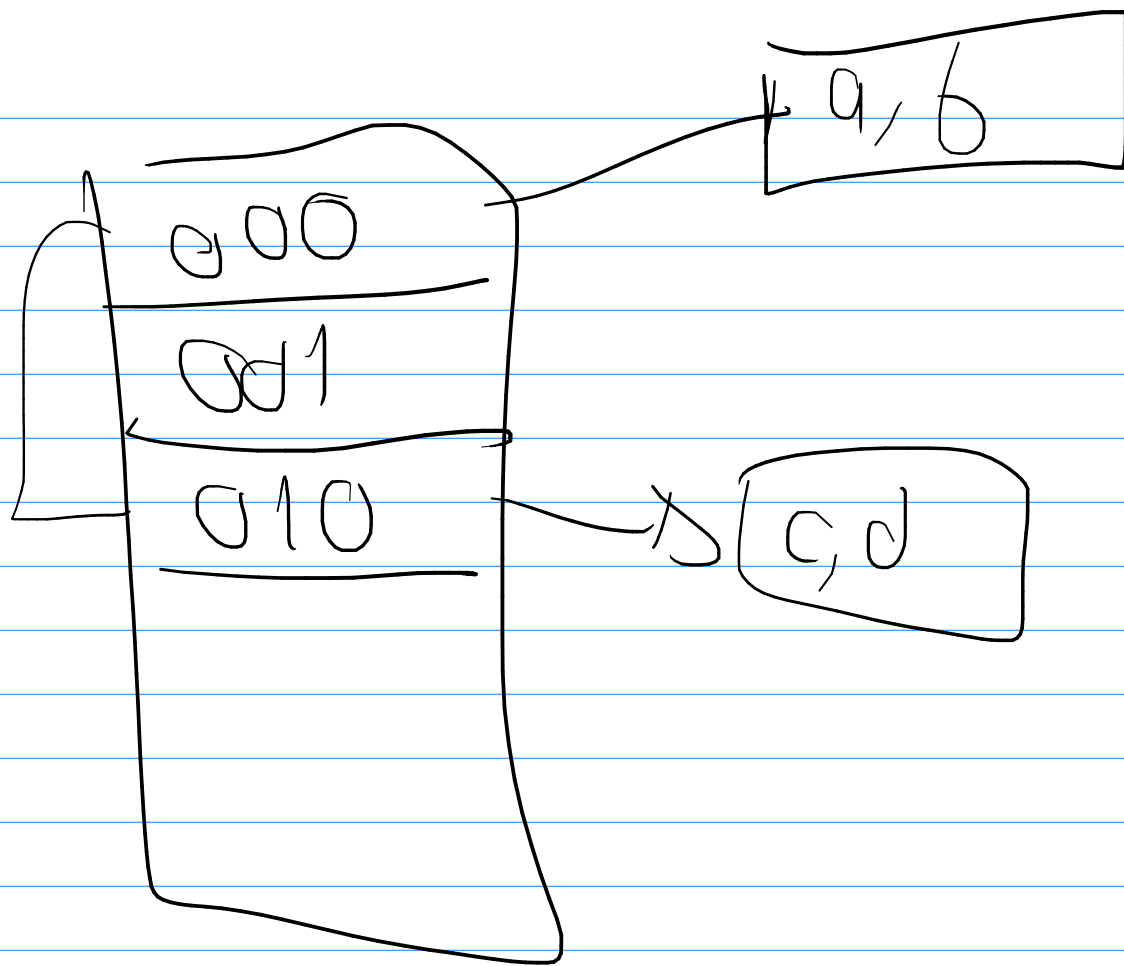
head =

head = 2



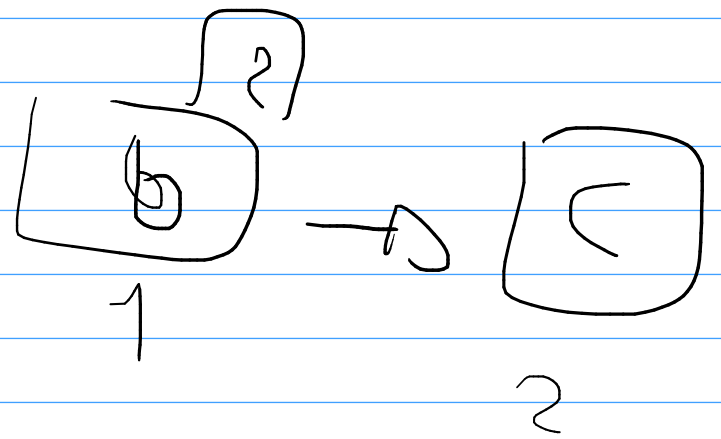
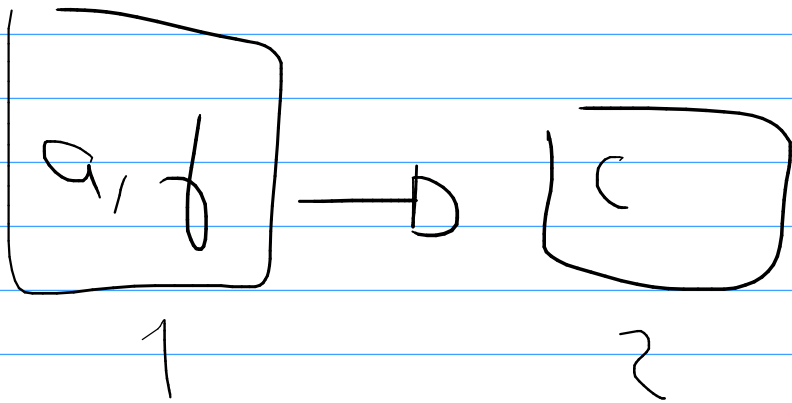
124





head = 2
free = -1

Removal of head = 1



```
----- 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
```