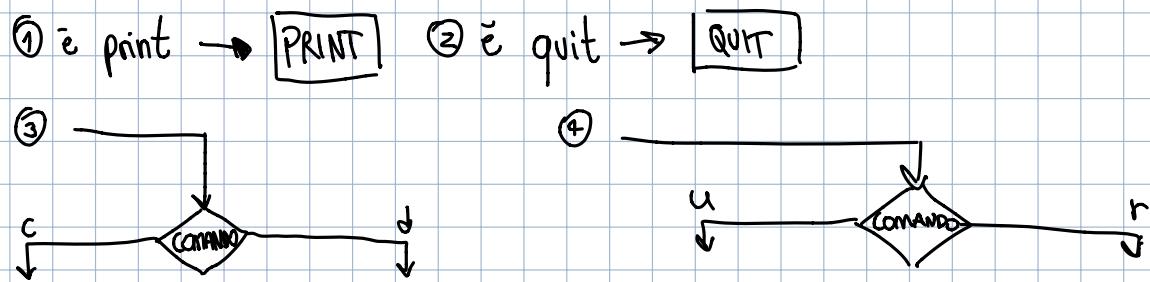


command → c, d salva nella cronologia il complementare  
 → u, r  
 → q, p esegue e basta

Flow : APPO → Aspetto input → COMANDO

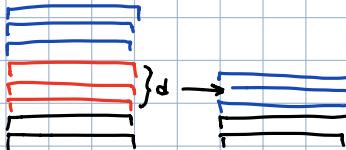
①  
②  
③  
④



## COMANDI COMPLEMENTARI

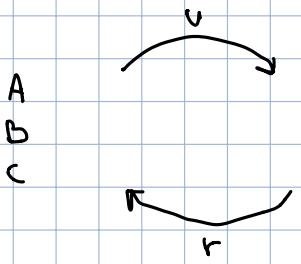
$i_1, i_2, d \rightarrow$  elimina le righe  $i_2$  in  $> i_1$  ①  
se  $i_1 >$  num. righe non manda nulla

① le righe vengono rimpiazzate → cassata



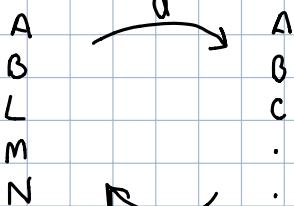
→ COMANDO COMPLEMENTARE → insert

1,3 c  
A  
B  
C

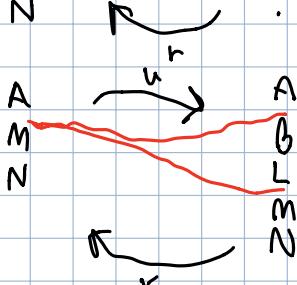


change: la riga non esiste  
↳ insert

3,5 c  
L  
M  
N



2,3d



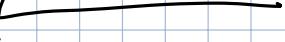
(a,b) delete  $\leftrightarrow$  insert (a)

## Searching (GetLine (int index))

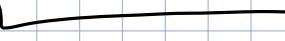
text.First



cursor.addr



text.last



P

From

# UNDO & REDO

UNDO ISTANCE zero  $\rightarrow$  len = 0  
 $x \mid \rightarrow$  set length to  $x$

① 1, 2 c  
CIAO  
AMICO

1, 1c CIAO  
2, 2c AMICO

①  
①

01 ①

② 1, 3 c  
Ciao  
amico  
mio

1, 1c Ciao  
2, 2c amico  
3, 3c mio

①  
②  
②

1, 1c CIAO  
2, 2c AMICO  
21

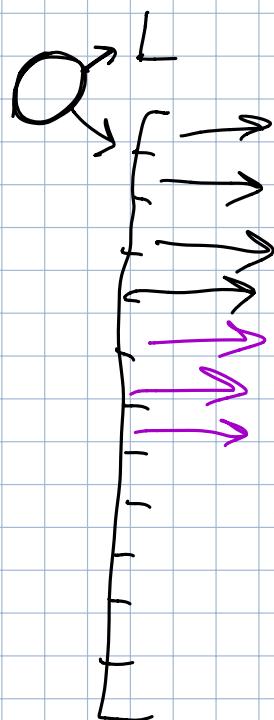
③ 2, 2 d  
1: Ciao  
2: mio  
L=2

2, 2d

③

2, 2 i amico

Hinsert  $\rightarrow$  len++



③\* 1, 2 d

2, 5c

r

u

A A  
B W  
C X  
D Y  
E Z

c ← code → c  
z ← loc → z  
4 ← site → 2  
"W", ← lines → 'B',  
"X", ← lines → 'C',  
"Y",  
"Z",  
5 ← setlen → 3

↓  
delete  
(shift exec)

4, 8c

r

u

A A  
B B  
C C  
D D  
E E  
F F  
G G  
H H

setl -  
loc  
=size - 1

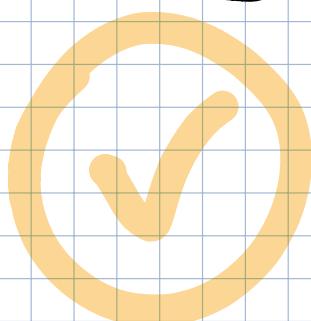
[d  
e  
f  
g  
h]

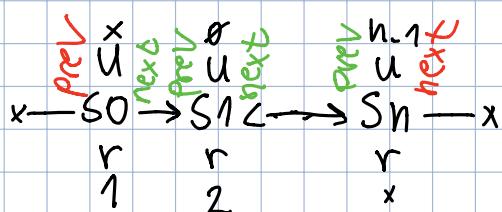
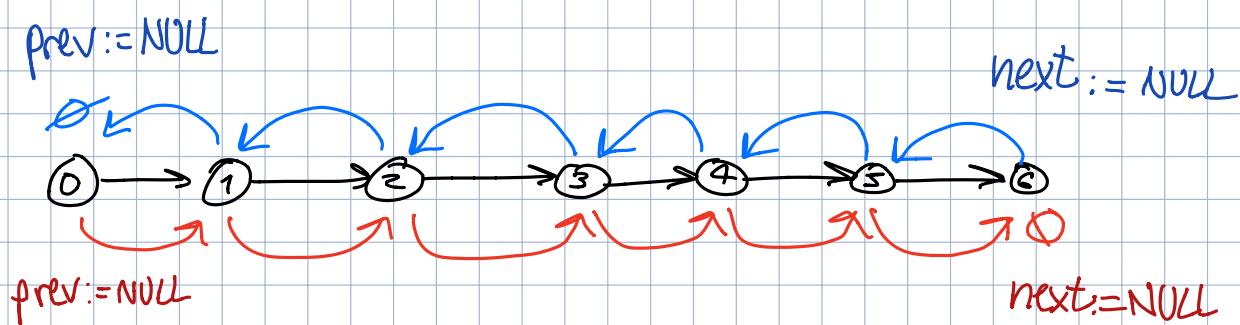
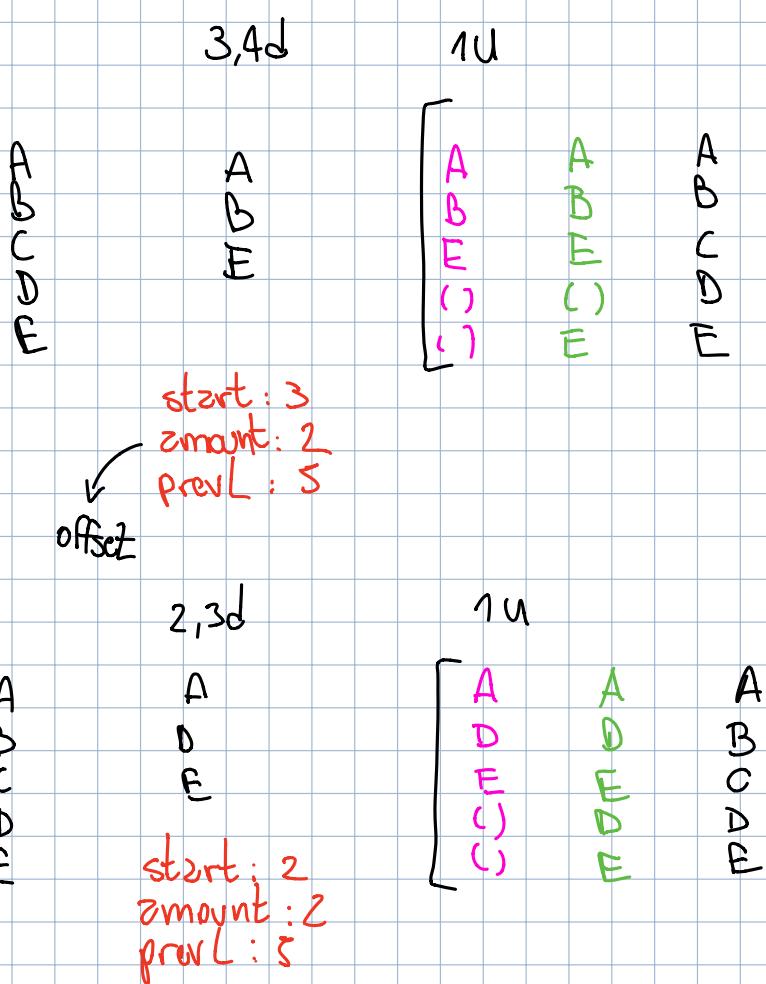
[D  
E  
F  
G]

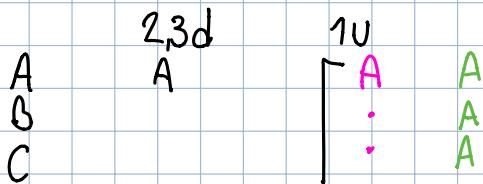
code  
loc  
size  
setl  
lines

setl -  
loc  
= size - 1

size = setl - loc + 1







SF: 2  
sm: 2  
ptl: 3

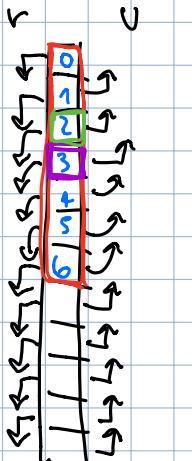
A  
B  
C  
D  
E

24d  
From: 2  
Offset: 3

A  
E  
C  
D  
NULL  
free  
free

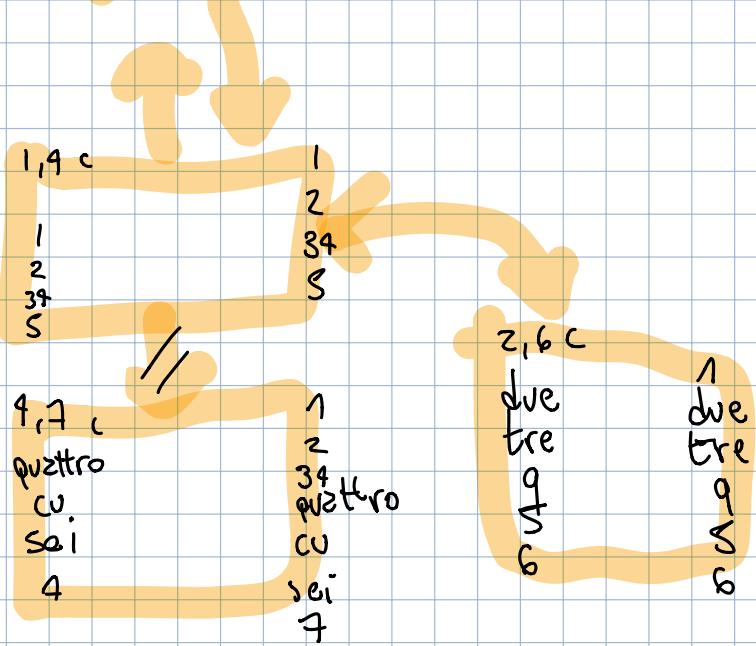


count = 8  
current = 6

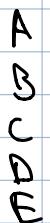


count = 5  
current = 5 4th

count = 4  
current = 2 3rd



4,6 d  
loc = 4

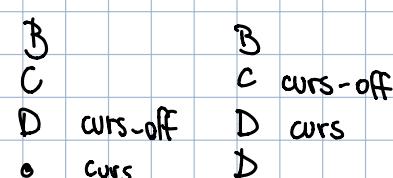
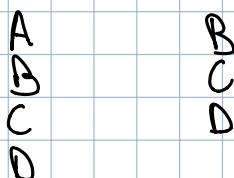


2,5d

A

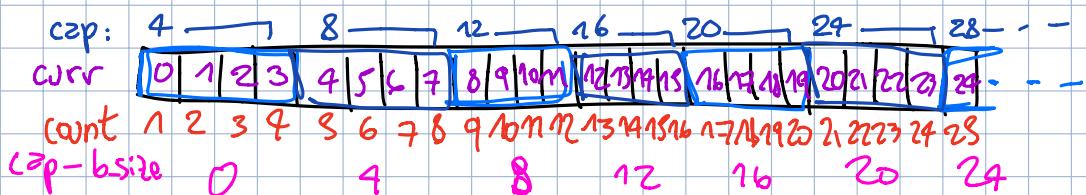
1U(2,5i)

A  
•  
•  
•  
•



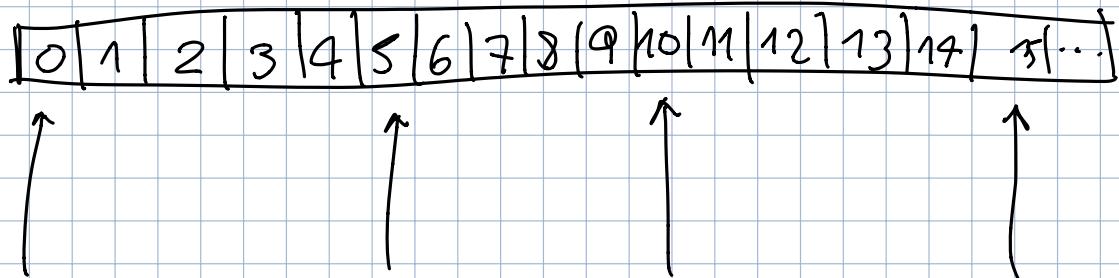
# OPERAZIONE: ACCELERARE ROLLINGBACK

(init)  $s\_cap = 0$       if  $s\_count == s\_cap \rightarrow$  ALLOC NEW BLOCK  
 $s\_count = 0$        $\rightarrow$   
 $s\_curr = 0$       else  $Count++$



s\_count ++

F = S



restore

setlen = t\_len  
size = t\_len  
lines = text (copied to)

\* if (count > 0 & count % F == 0)

in **UpdateHistory()** {

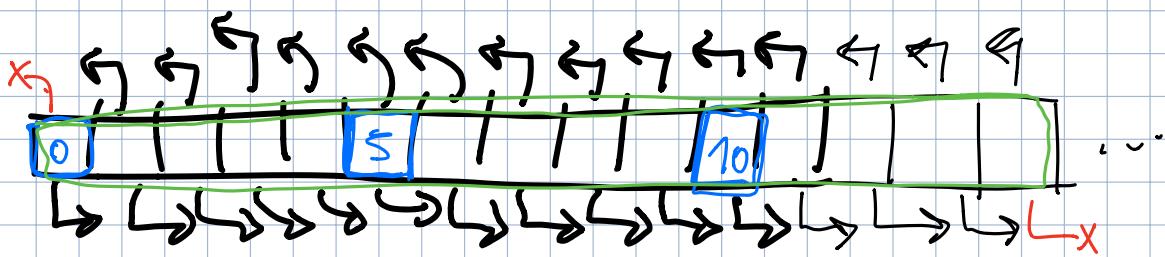
| :  
| :  
| :  
| :

h[count - 1].redo = NULL  
h[count - 1].undo = NULL

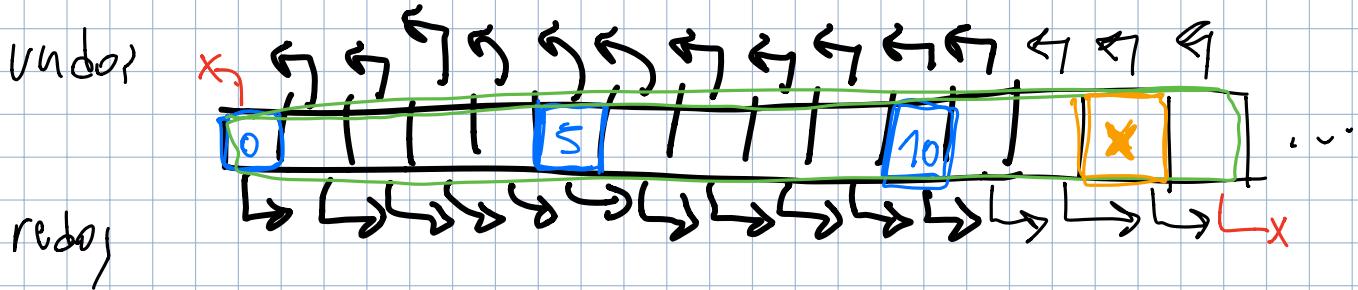
}

F = S

undo}



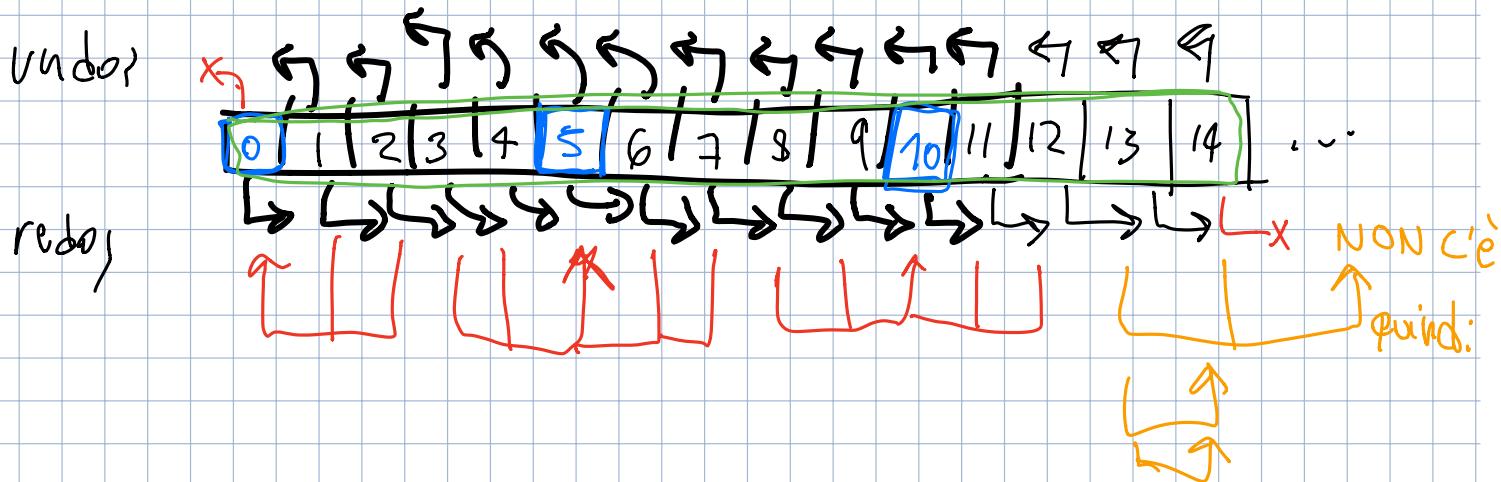
redo}



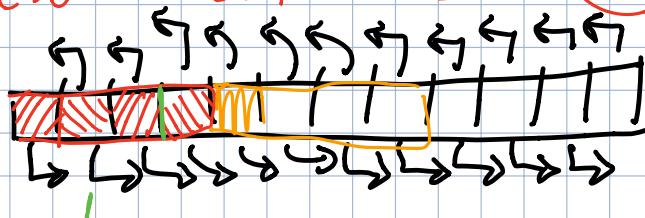
$\times$  Undo  $\rightarrow$  trovo restore + vicino  $\rightarrow$  mi muovo di e calcolo stato di arrivo quanto manca

$$\text{targetState} = \text{currentState} - \#U$$

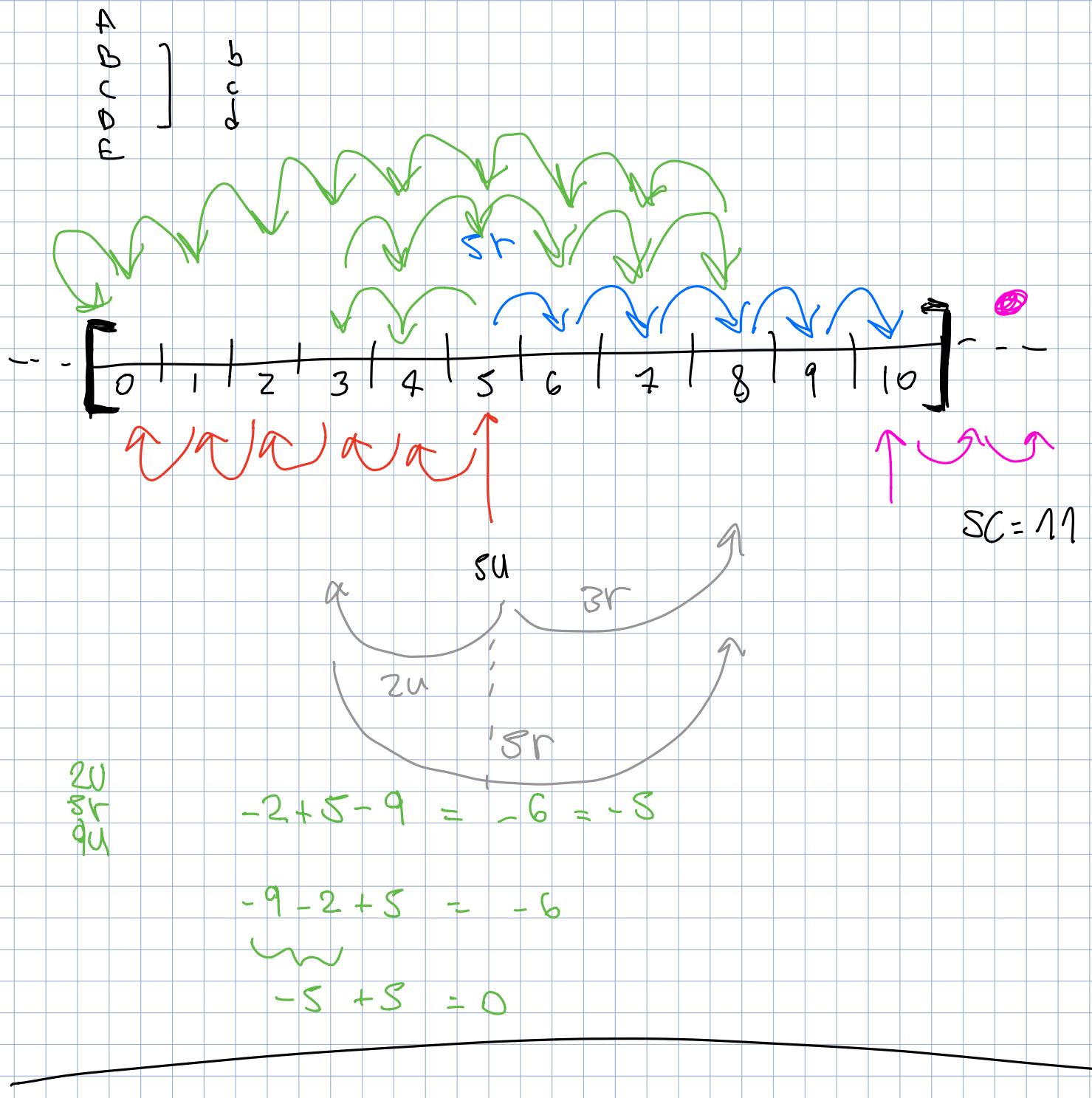
Se  $tS \leq 0 \rightarrow \text{closestRestore} = 0 ; tS = 0 ;$



$$\begin{array}{l} S=0 \\ N=0 \\ C=0 \end{array} \rightarrow \begin{array}{l} S=4 \\ N=1 \\ C=0 \end{array} \rightarrow \begin{array}{l} S=4 \\ N=2 \\ C=1 \end{array} \rightarrow \begin{array}{l} S=4 \\ N=3 \\ C=2 \end{array} \rightarrow \begin{array}{l} S=4 \\ N=4 \\ C=3 \end{array} \rightarrow \begin{array}{l} S=6 \\ N=5 \\ C=4 \end{array}$$

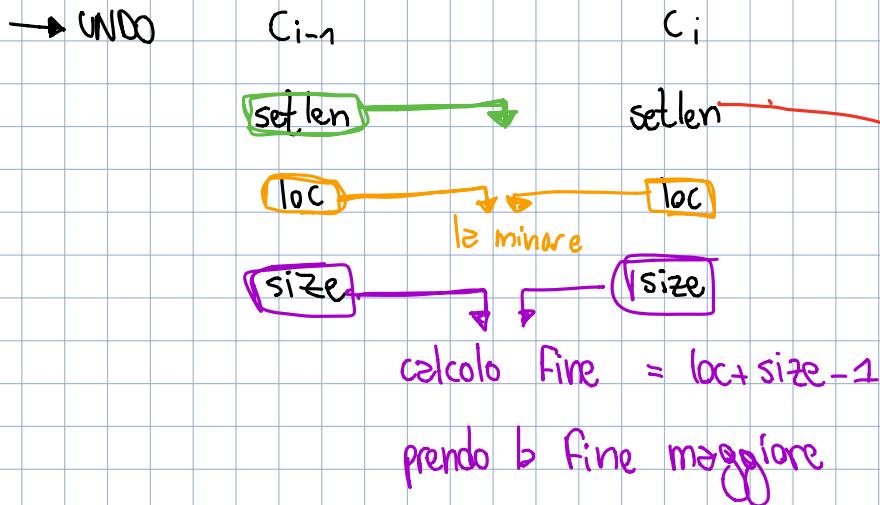


$$\begin{array}{l} S=8 \\ N=5 \\ C=1 \end{array}$$



OPERAZIONE ROLLING BACK 2: GLI ULTIMI 2 PUNTI

**IDEA:** anziché eseguire tutte le undo/redo delle change, se ci sono più edit(c) di fila sovrapporre le ed eseguirne una alla fine.



ImprintChange(A, B); // Imprints B onto A

• UNDO  
C.setlen = A.setlen  
C.loc = min(A.loc, B.loc)

$$\begin{aligned} \text{to\_A} &= A.\text{loc} + A.\text{size} - 1 \\ \text{to\_B} &= B.\text{loc} + B.\text{size} - 1 \\ \text{to\_C} &= \max(\text{to\_A}, \text{to\_B}) \\ \text{C.size} &= \min(C.\text{setlen}, \text{to\_C}) - C.\text{loc} + 1 \end{aligned} \quad ] \quad \text{to\_C} = \max(A.\text{loc} + A.\text{size}, B.\text{loc} + B.\text{size}) - 1$$

C → lines = realloc di C.size × sizeof..

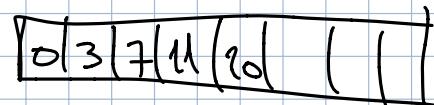
for(int i = 0; i < C.size; i++) {

    if(C.loc + i > A.loc + i && C.loc + i < A.loc + A.size)  
        C.lines[i] = A.lines[i];

}

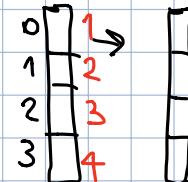
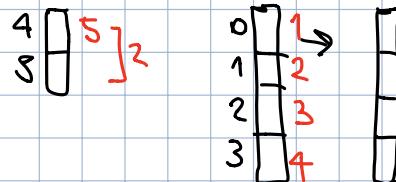
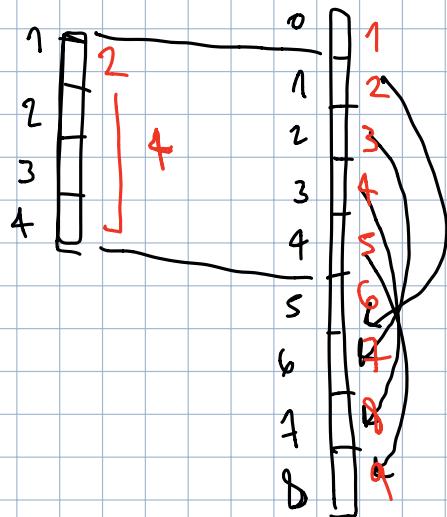
# SNAPSHOT, QUESTION MARC?

state  
 ↗ undo  
 ↘ redo  
 ↘ closestSnapshot

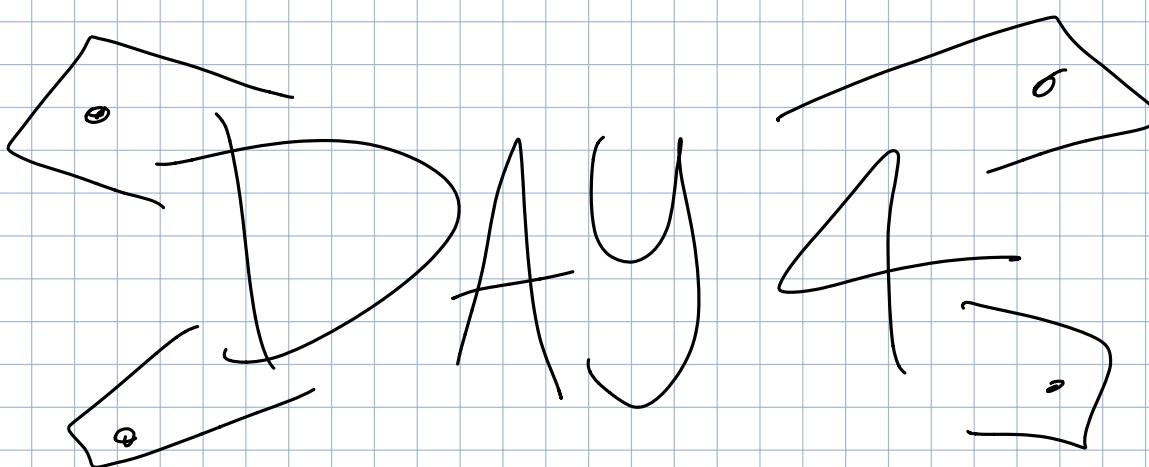
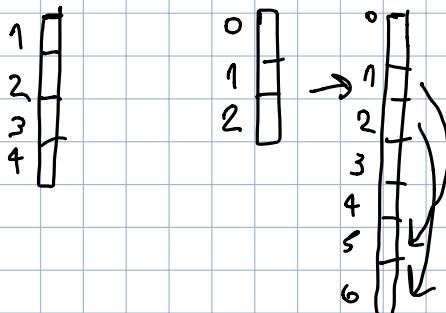


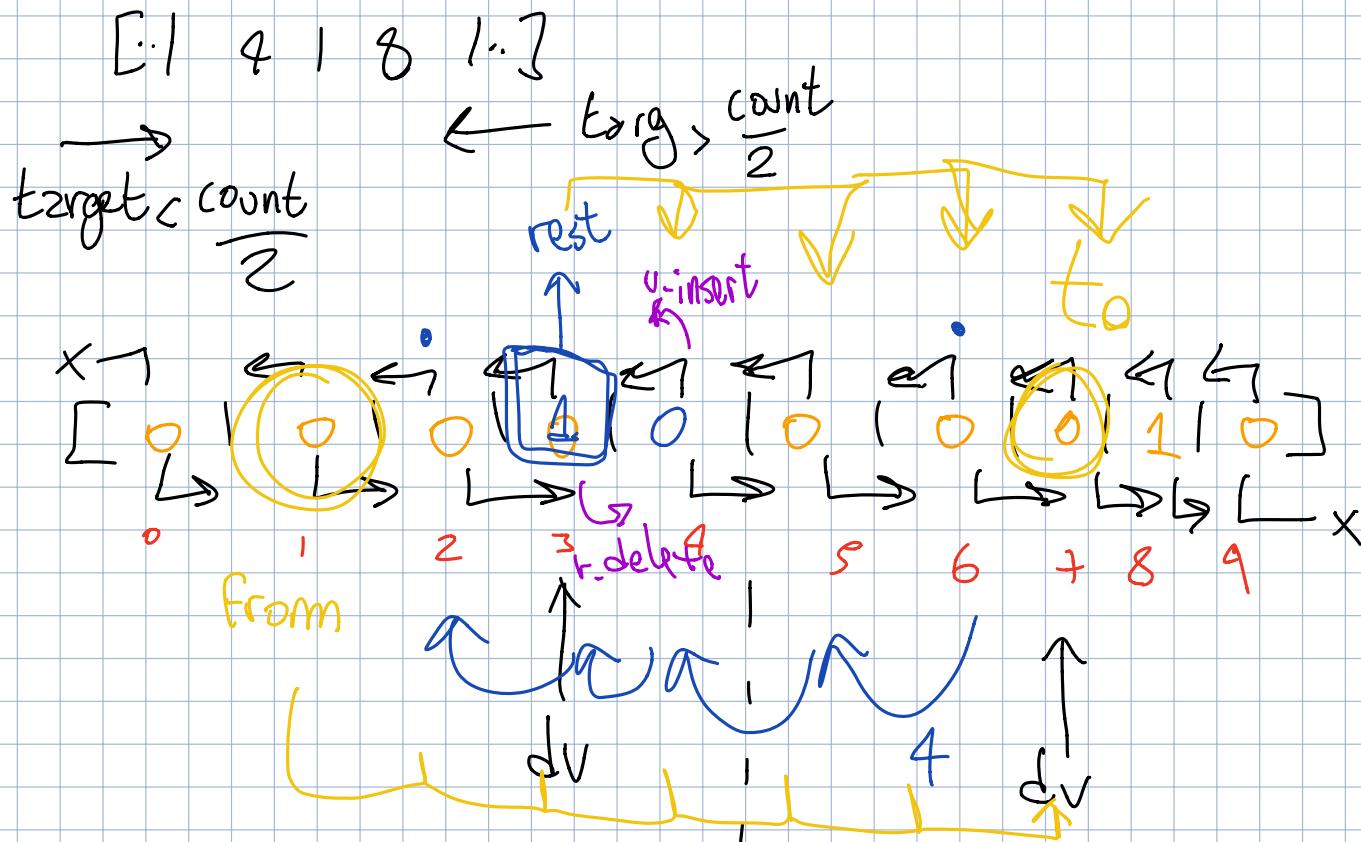
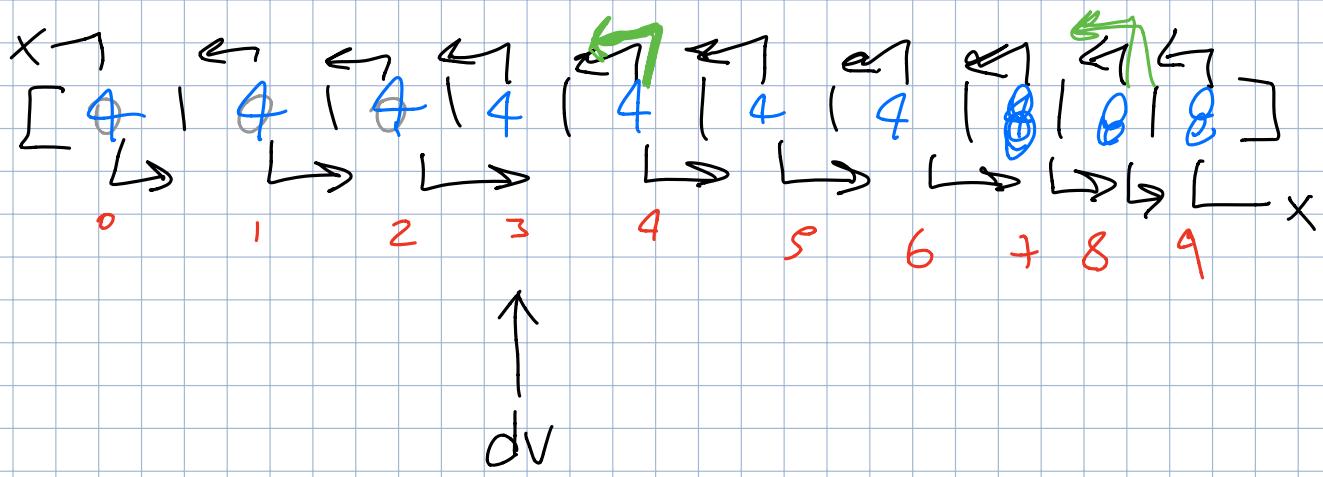
- PARTO DA UNA FINE
- FINO A CHE  $i > loc + off$

$$t[i] = t[i - off]$$

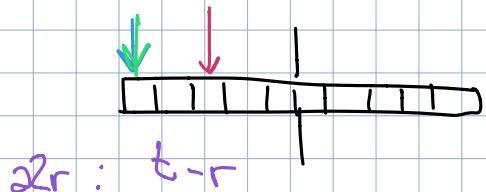


$$\dots i = 5 - loc$$



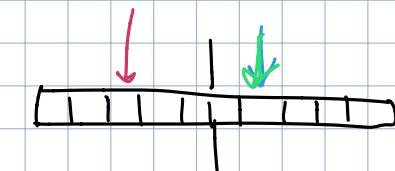


## TARGET CURRENT RESTORE

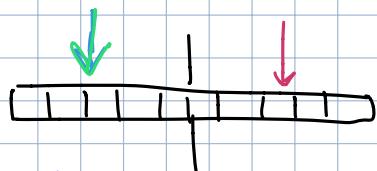


$dt$ : dist(curr, target)  
 $dr$ : dist(rest, target)

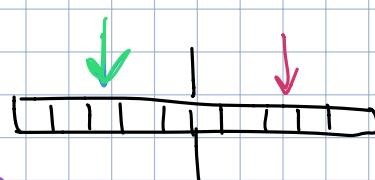
$$2r : t-r$$



22n : t-+



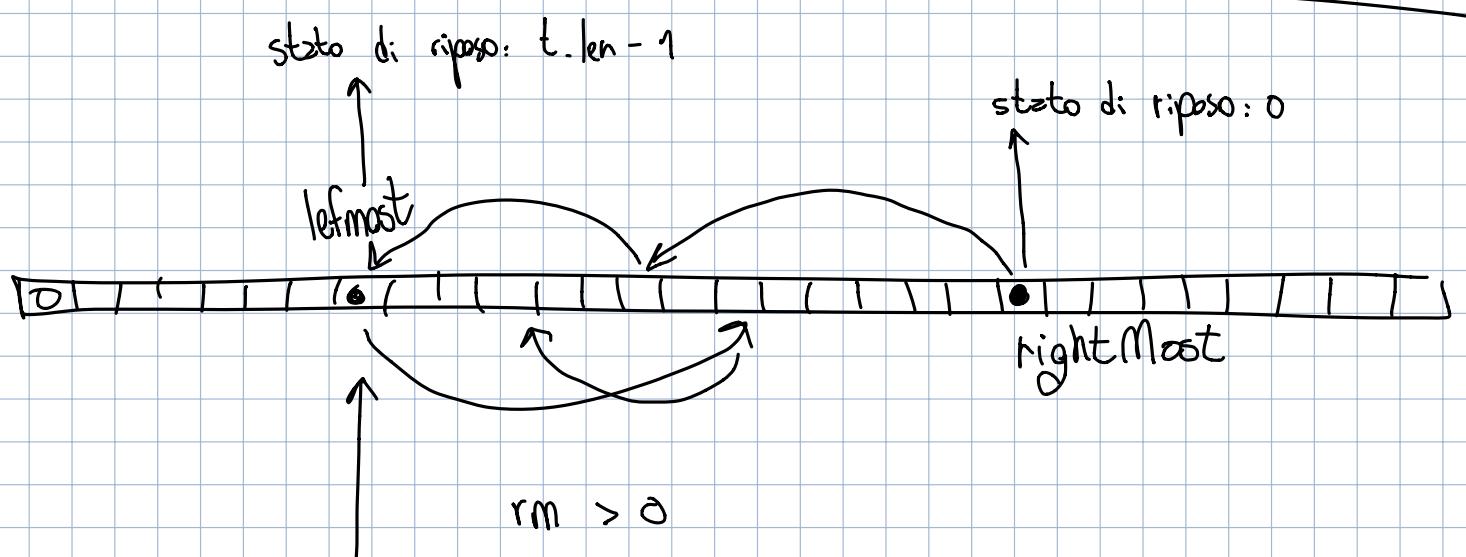
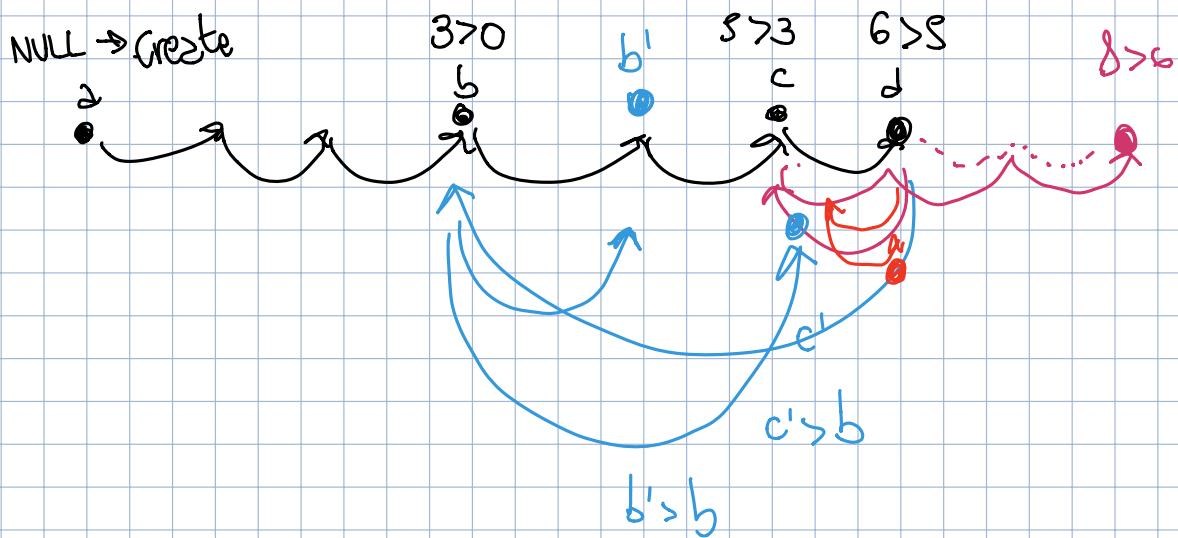
$$2^2r : t-r$$



$$22r = E - r$$

[0 1 2 3 4 5 6 7 8]

9 / 2 - 4



~~- se lm = 0  $\rightarrow$  Solo Setlen(0)~~

closest multiple to (target)

$d = target;$

$c = 0 \quad // closest$

for ( $i = 0; i < statecount - 1 \times F; i++$ ) {  
    if ( $\text{abs}(i \times F - target) < \text{abs}(c - target)$ ) {

$c = i \times F$

$d = \text{abs}(c - target)$

}

$$c - target \quad |target| \underbrace{|c|}_{\text{min}}$$