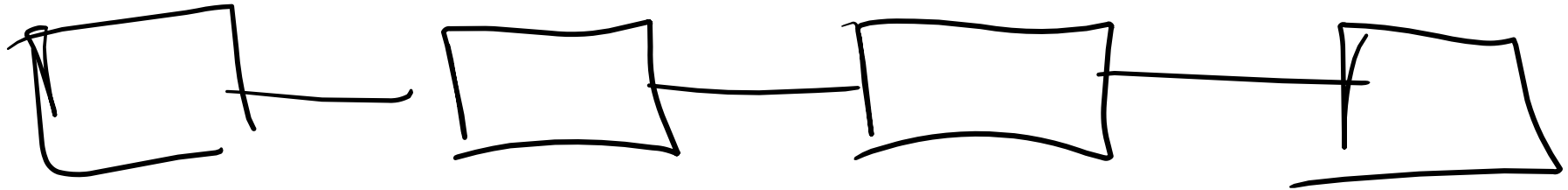


code

# queue

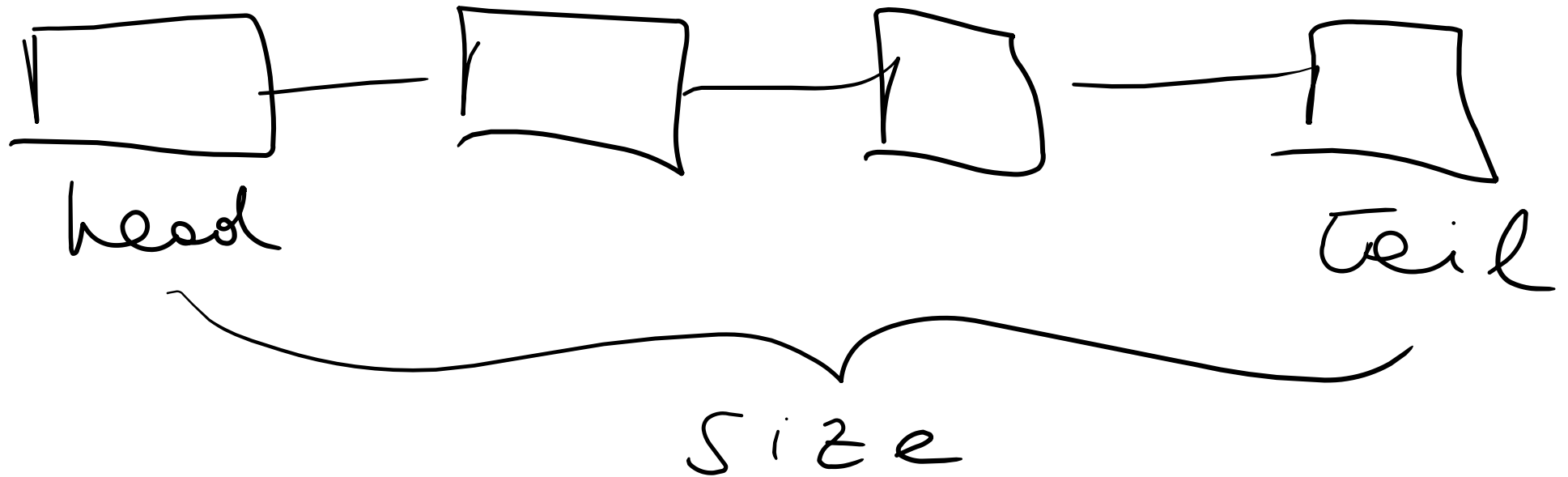
- Una serie di elementi uno dopo l'altro che verranno serviti in ordine di arrivo



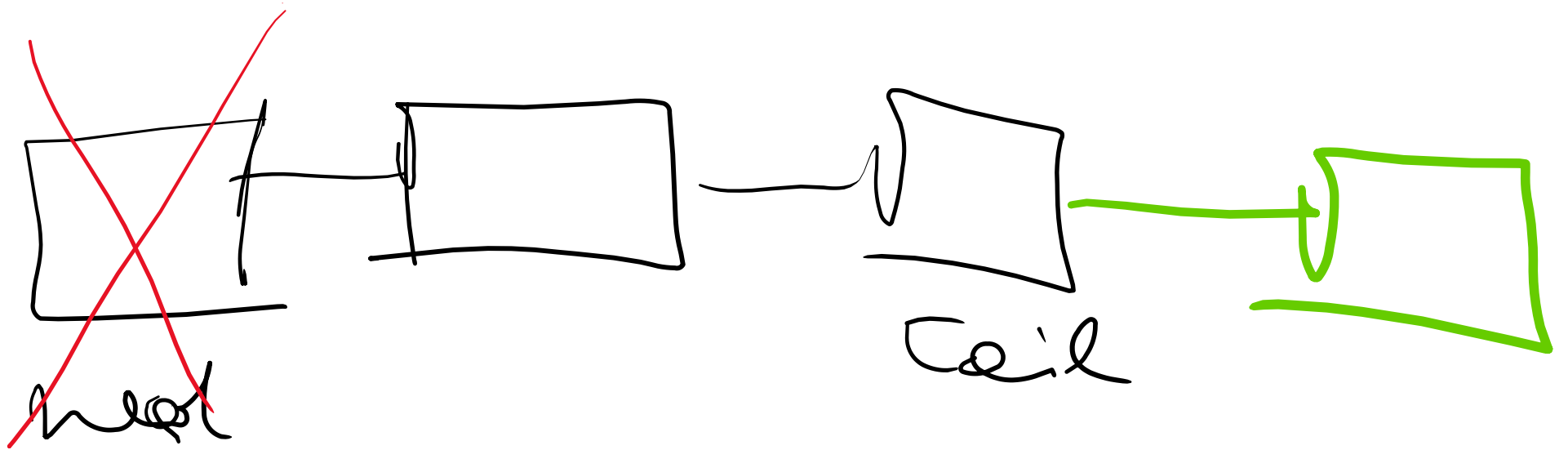
FIRST IN FIRST OUT

FIFO

queue



queue



insert Teil

remove Head

is Empty()



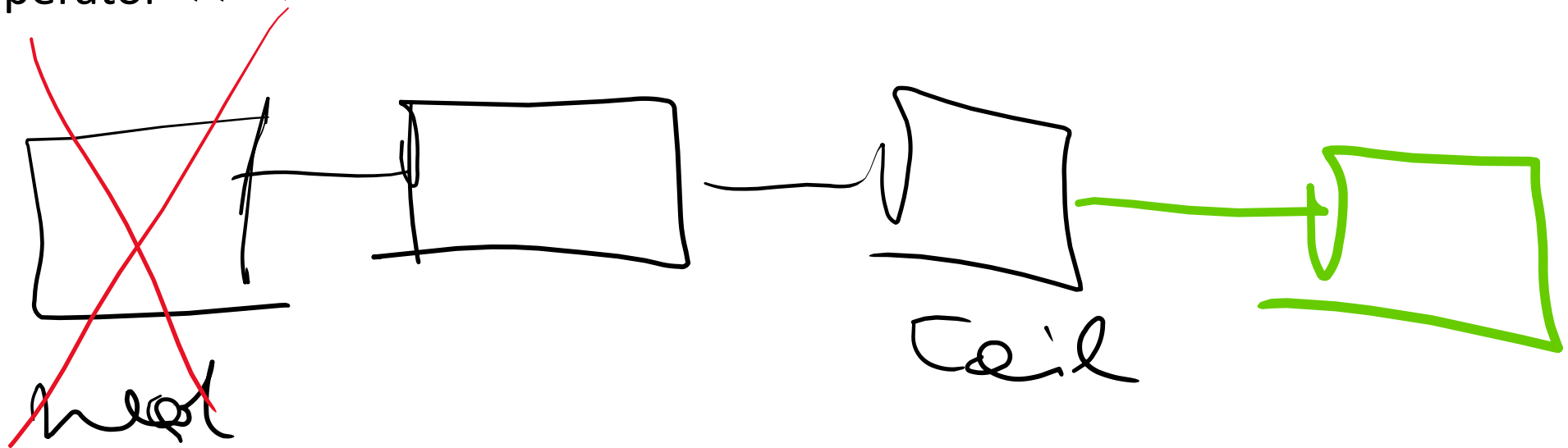
ENQUEUE



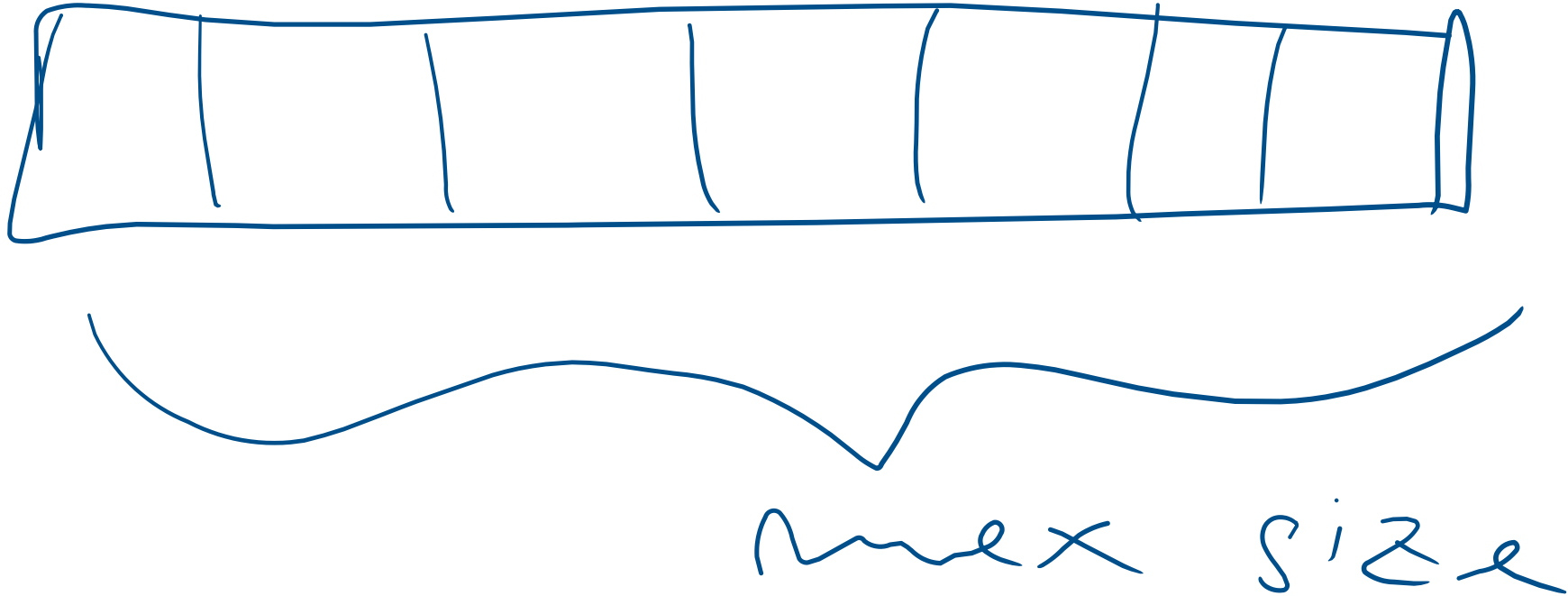
DEQUEUE

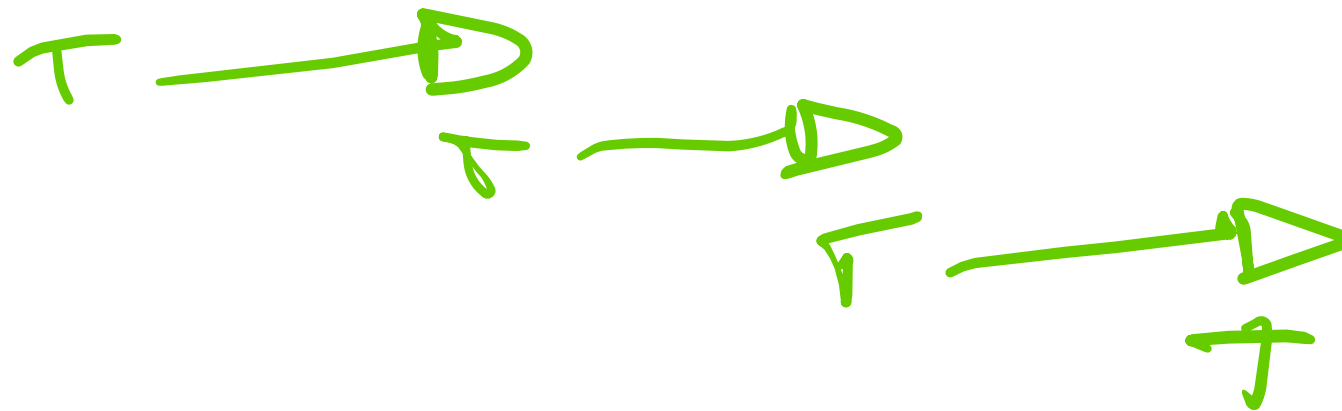
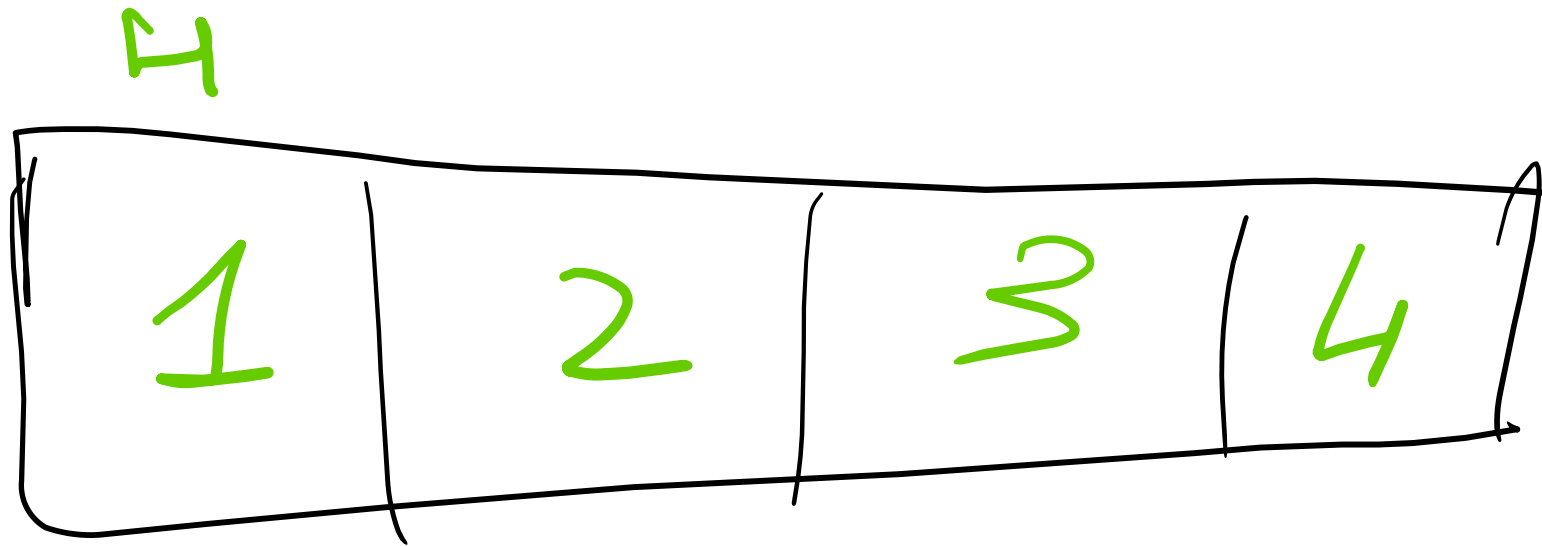
# Implementazione con DLList

- Estendere la classe DLList per implementare i metodi di
  - Enqueue
  - Dequeue
  - isEmpty
  - Operator <<



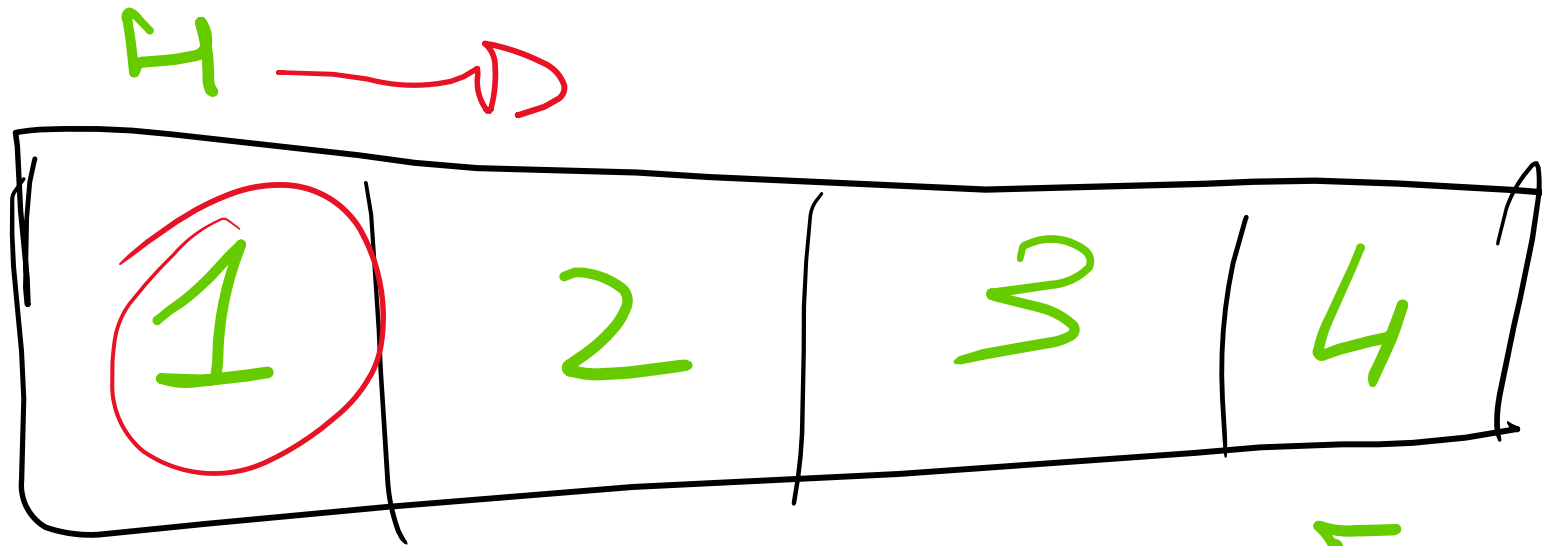
# Implementazione statica





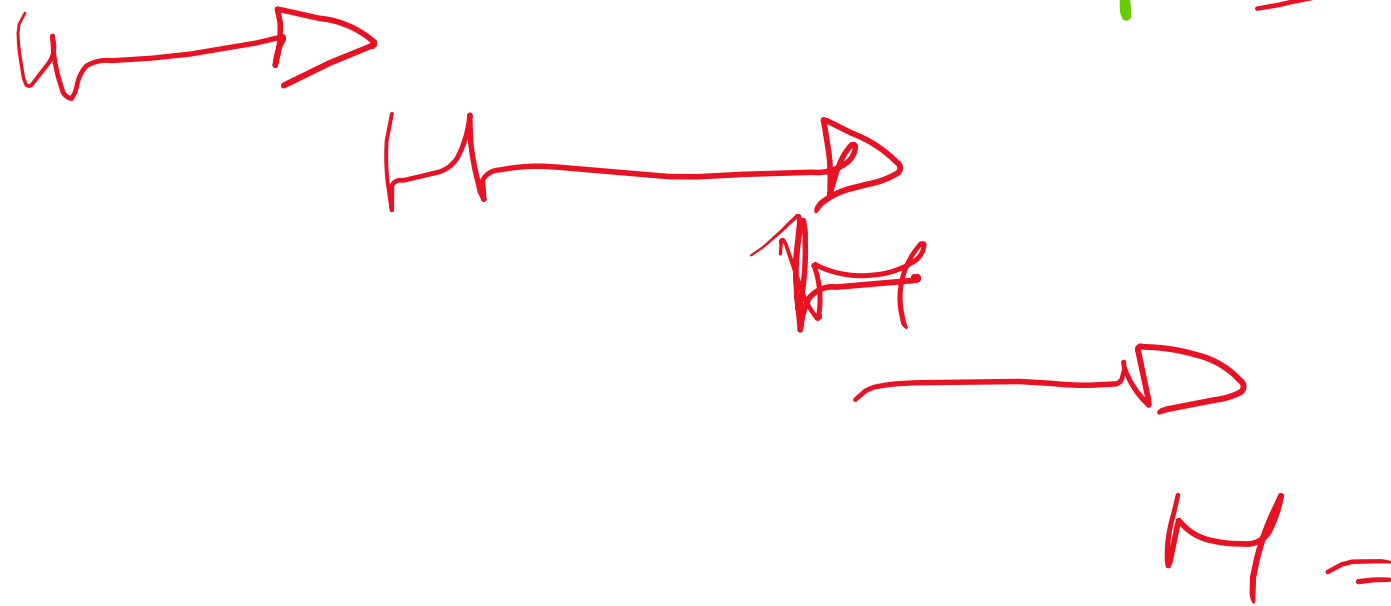
head = -1  
Tail = -1  
Size = 4

E(1)  
E(2)  
E(3)  
E(4)

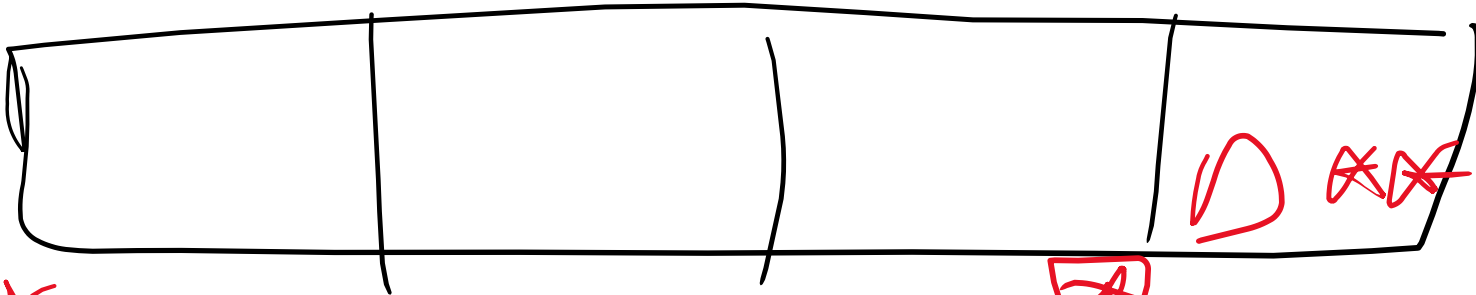


→ D ()  
 D ()  
 D ()  
 D ()

head = 0  
 Tail = 3  
 Size = 4



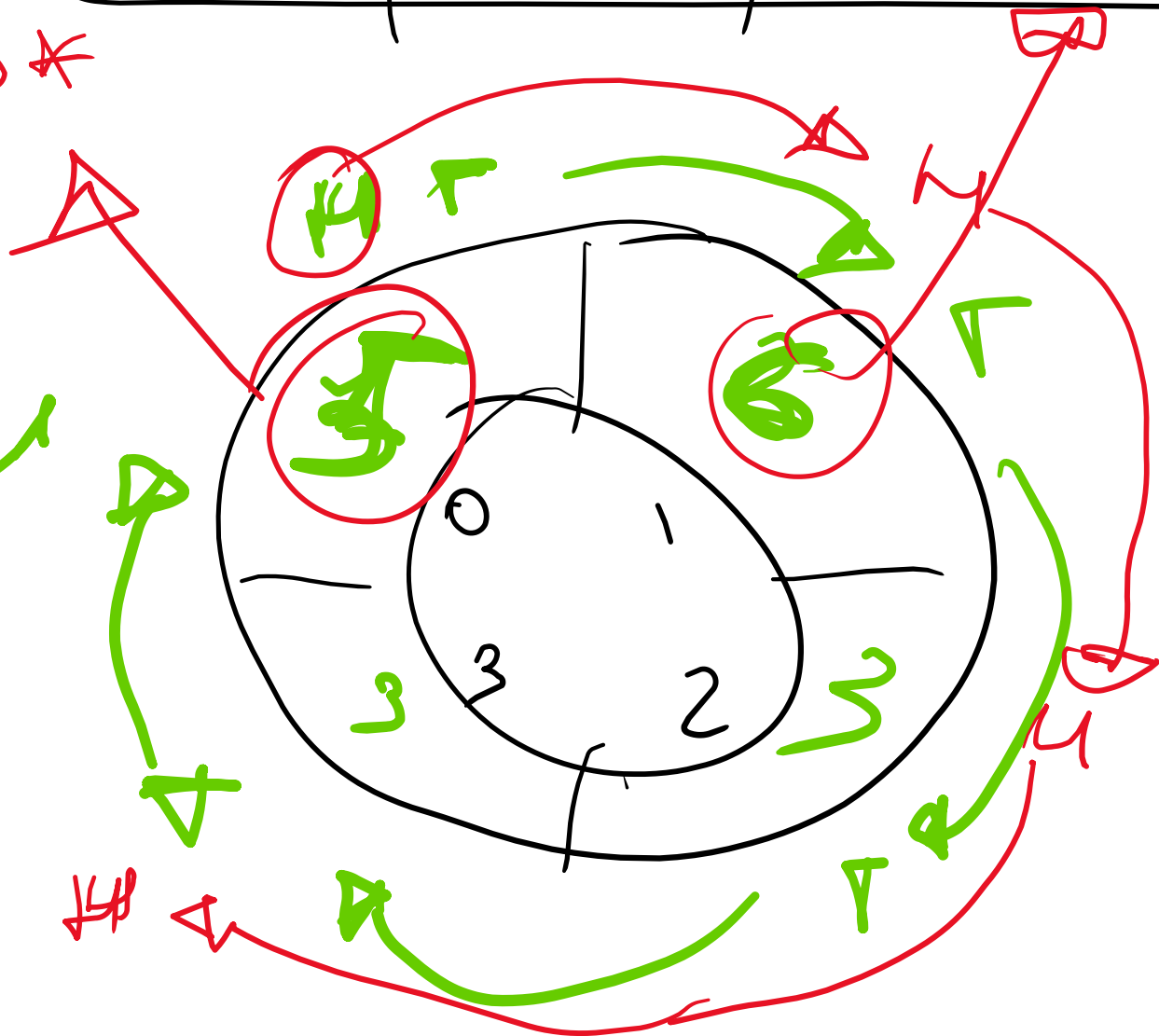




$D^*$

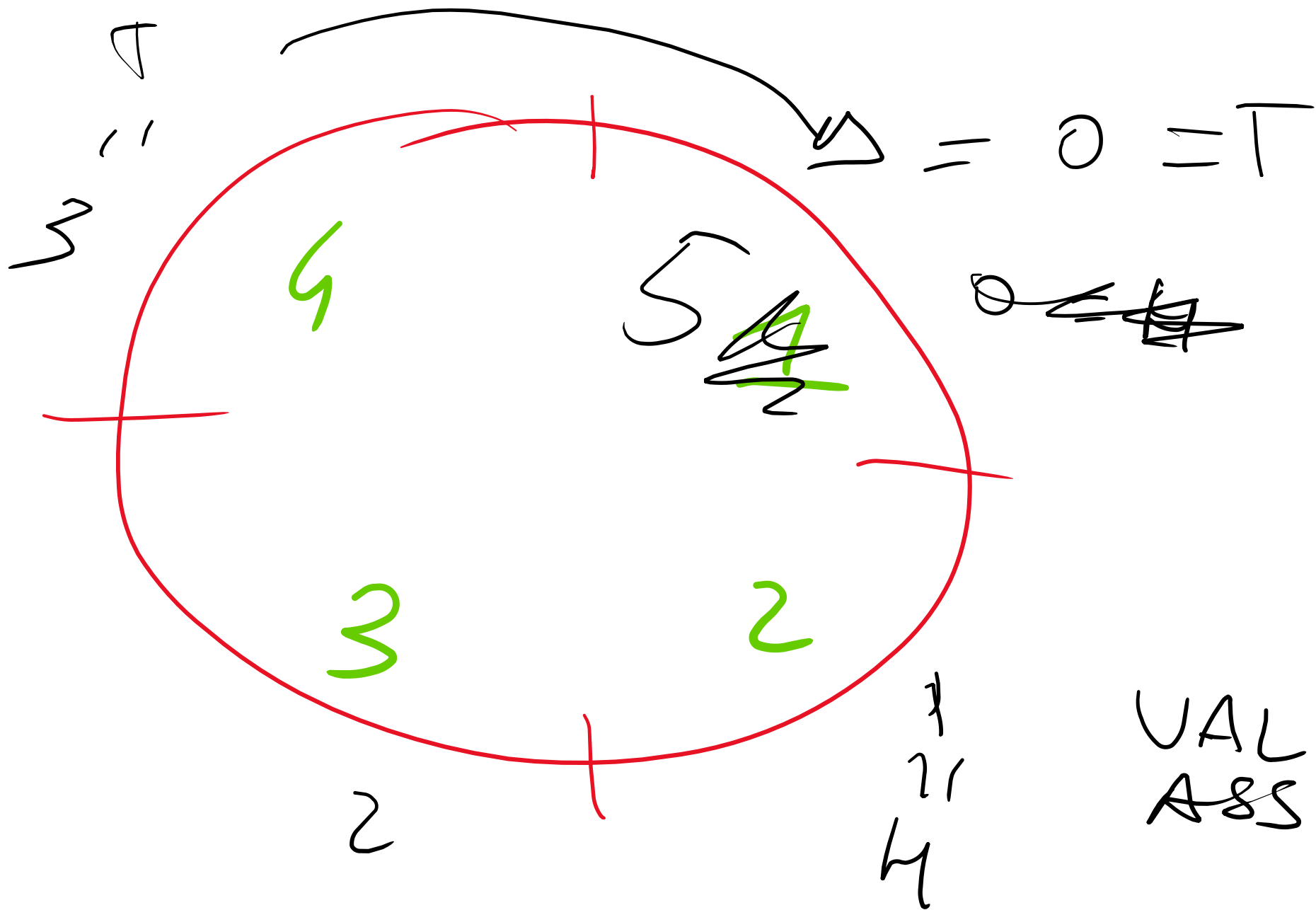
$\tau$

$\tau = 3$   
 $H = 3$



$E(1)$	$D$
$E(2)$	$D$
$E(3)$	$D$
$E(4)$	$D$

$E(5)$
$E(6)$
$D^*$
$D^{**}$



- Studente
  - Nome
  - Cognome
  - Età
  - Media

- Simulare un appello con una coda