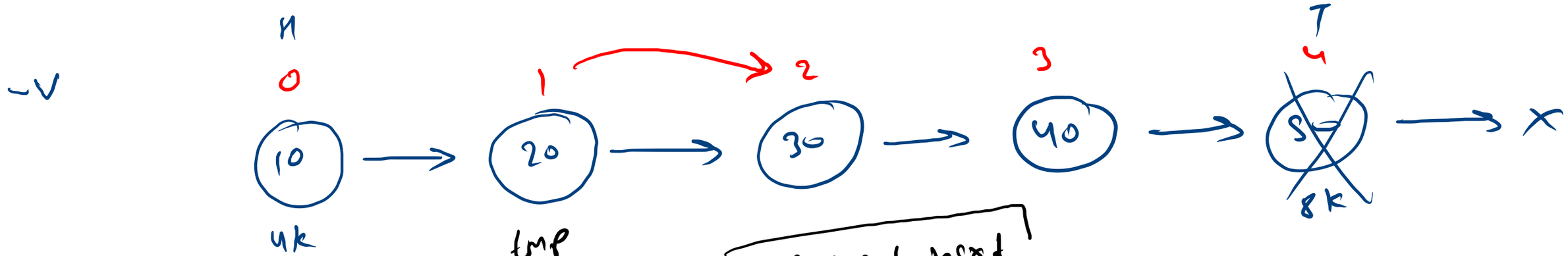
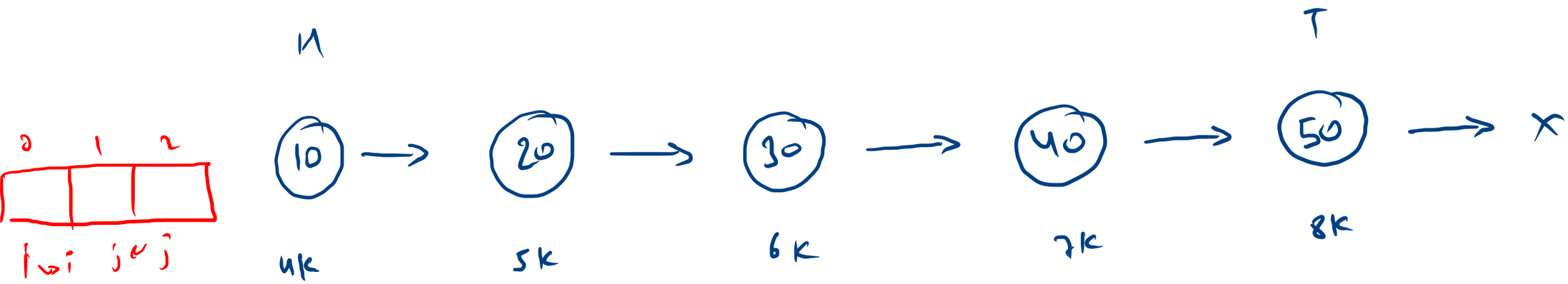


idx: 2

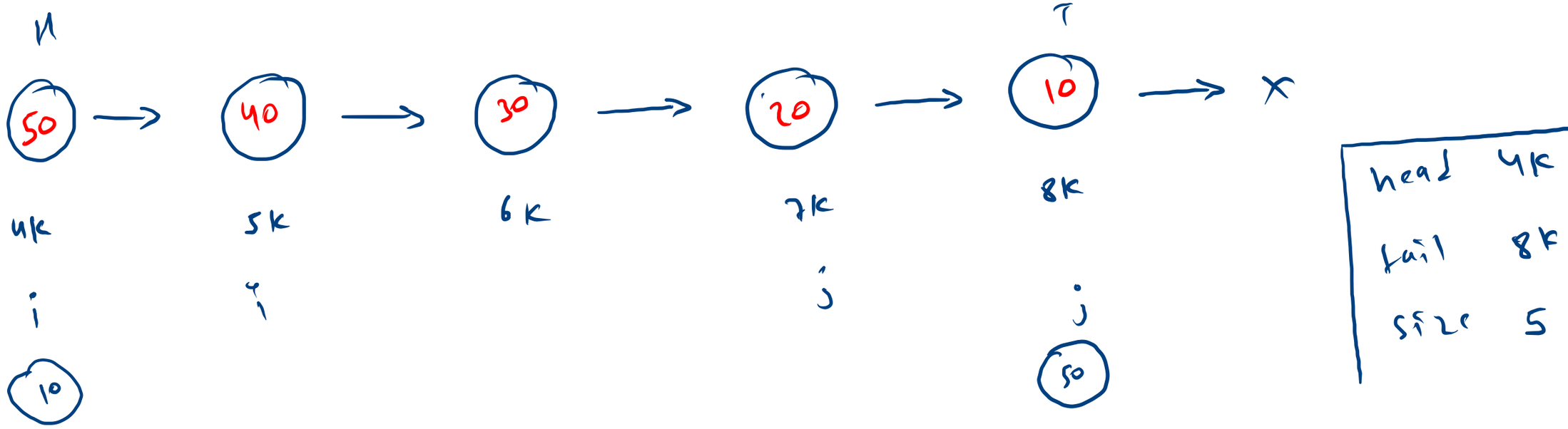
Head	uk
Tail	8k
size	5

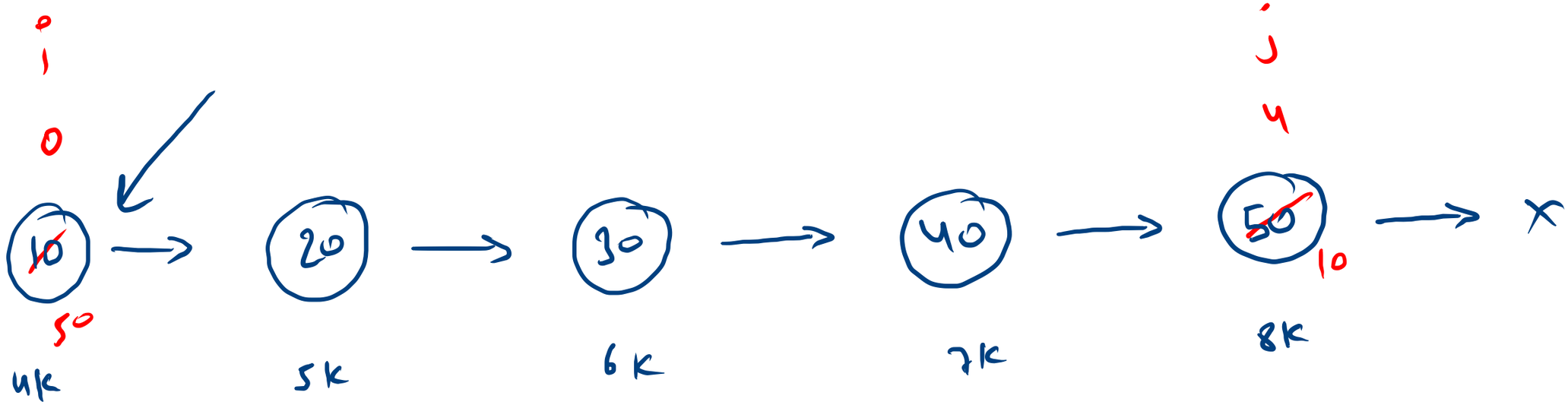


getvalue At (idx)

Iteration ✓

Recursion ✗





i = 0

j = size-1

Node a = getNode (i); (4K)

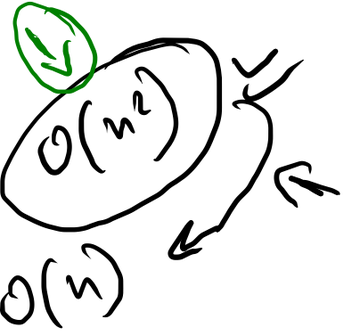
Node b = getNode (j); (8K)

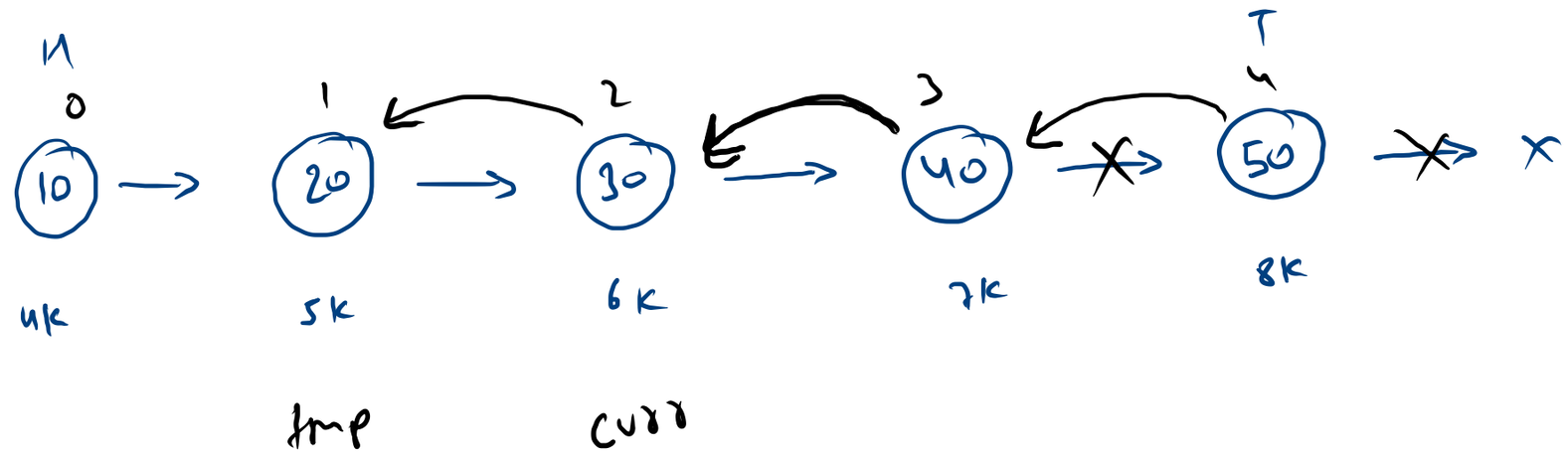
int tmp = a.data

a.data = b.data;

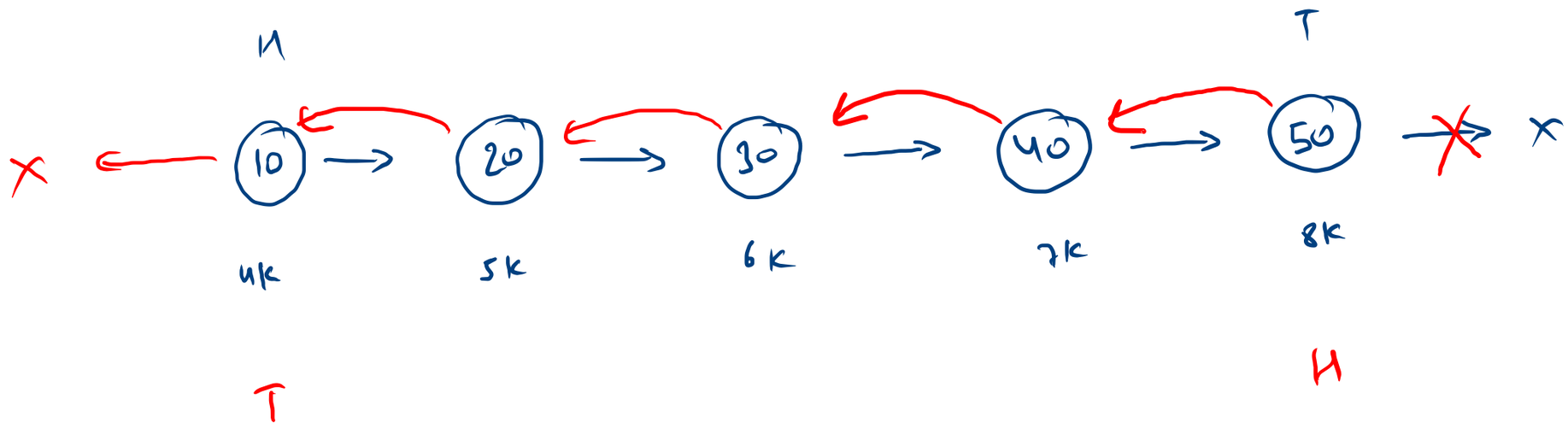
b.data = tmp;

tmp. data = 5

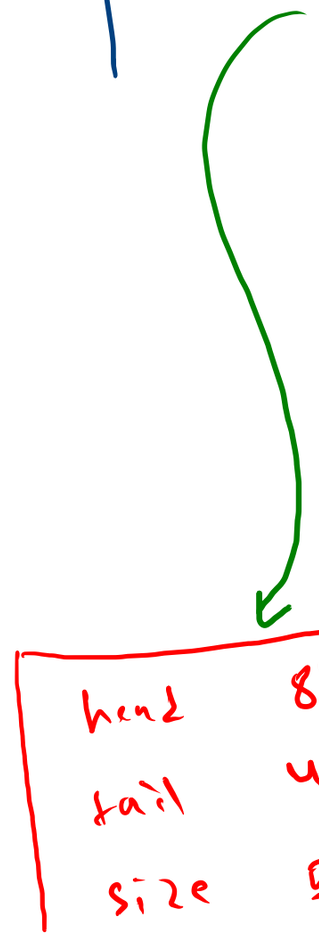




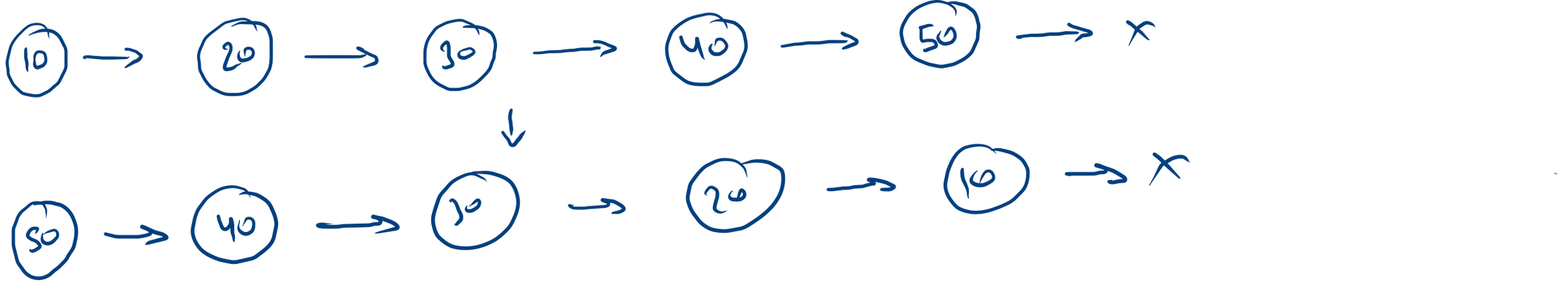
head 4K
tail 8K
size 5

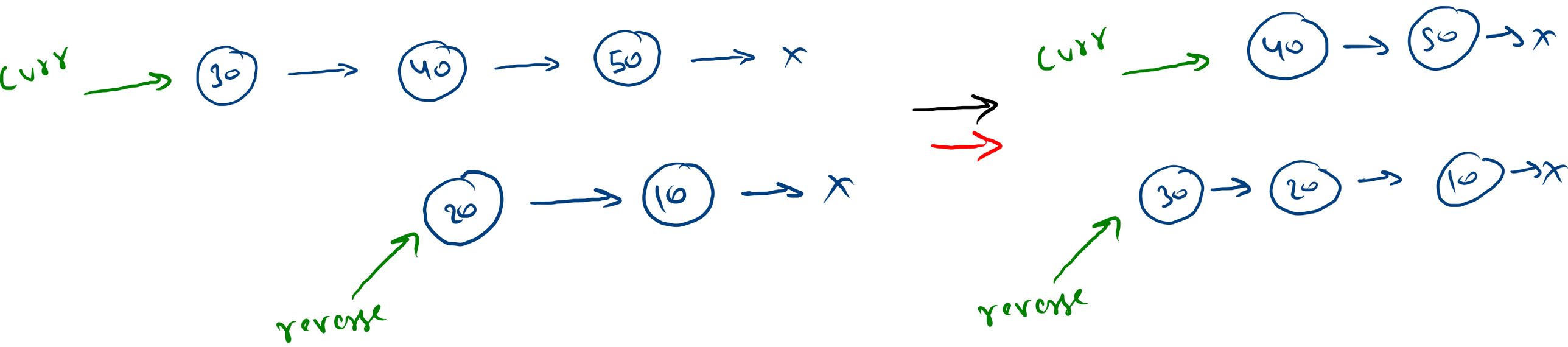


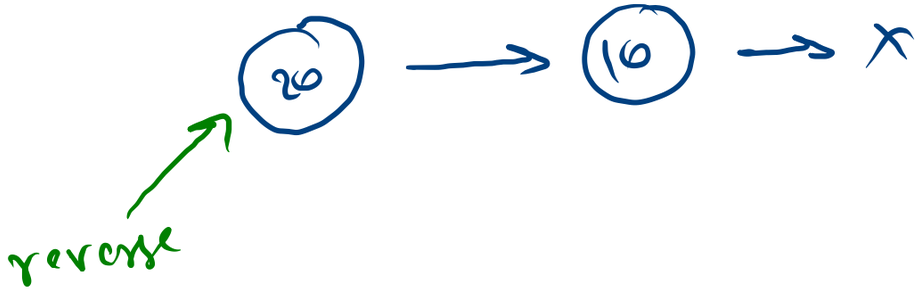
H

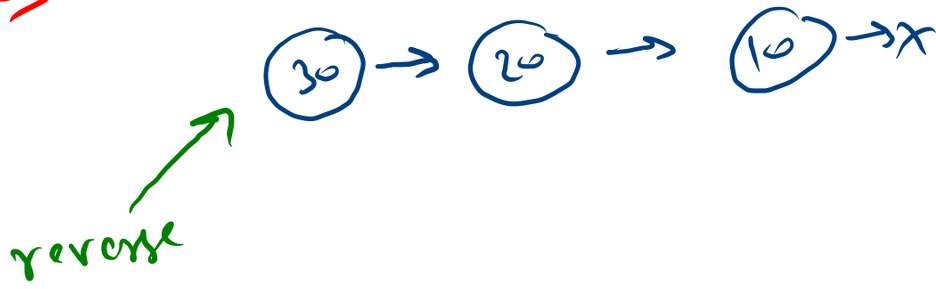


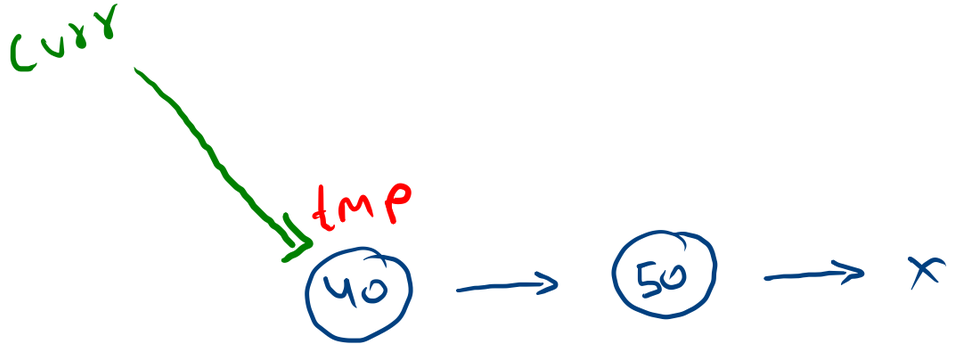
head 8K
tail 4K
size 5


10
20
30
40
50
X
50
40
30
20
10
X
30
20
10


curr
30
40
50
X
curr
40
50
X
20
10
X
reverse
30
20
10
X
reverse





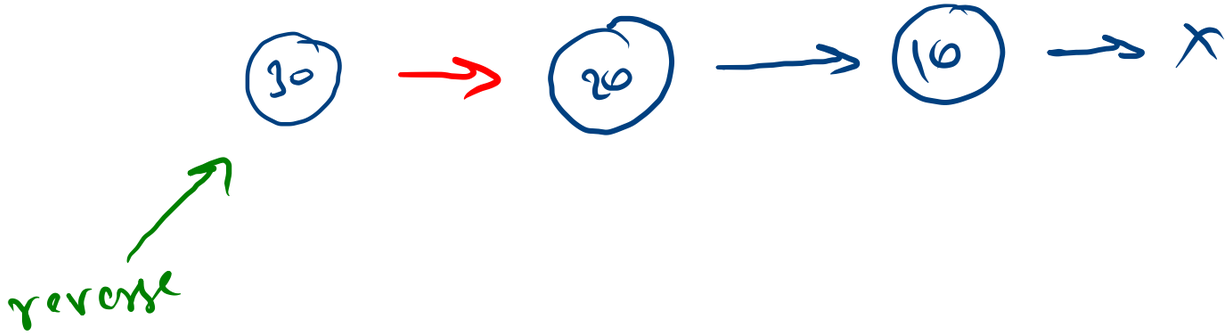


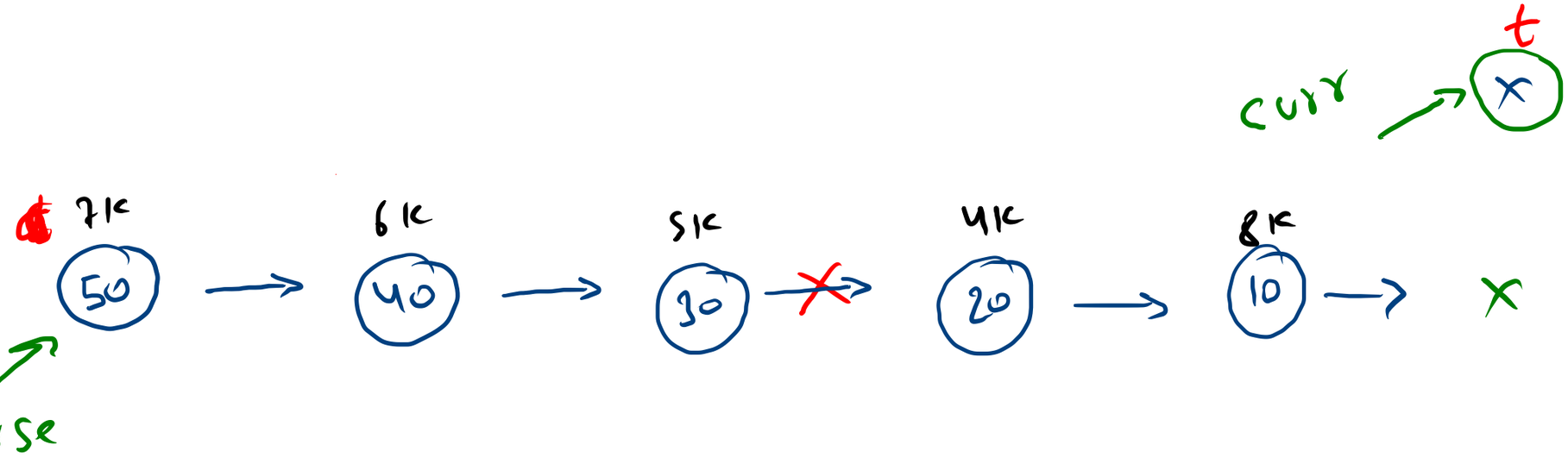
tmp = curr.next

curr.next = reverse

reverse = curr

curr = tmp






```
while( curr != null ){
    tmp = curr.next
    curr.next = reverse
    reverse = curr
    curr = tmp
}
```

curr ~~4K~~ ~~5K~~ ~~6K~~ ~~7K~~ X

reverse ~~8K~~ ~~4K~~ ~~5K~~ ~~6K~~ 7K

tmp ~~5K~~ ~~6K~~ ~~7K~~ X

curr = head

reverse = null

list.size()

list

push → addFirst

pop → removeFirst

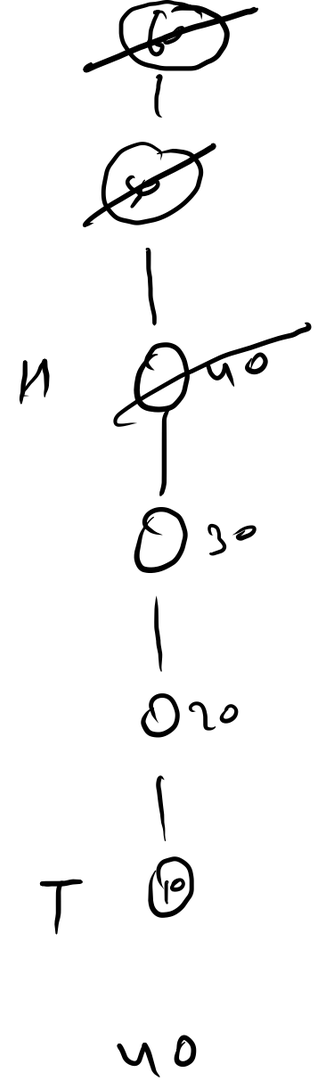
push addFirst
      addLast

pop   removeFirst
      removeLast

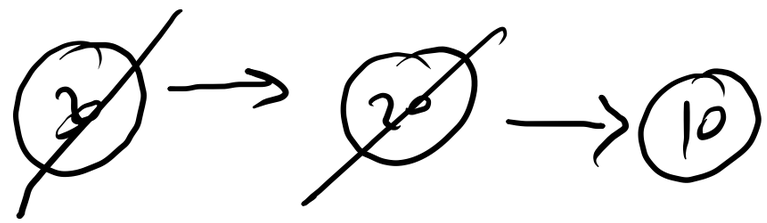
top   peekFirst
      peekLast

2 | add → addLast
  | remove → removeFirst

S | push
  | pop

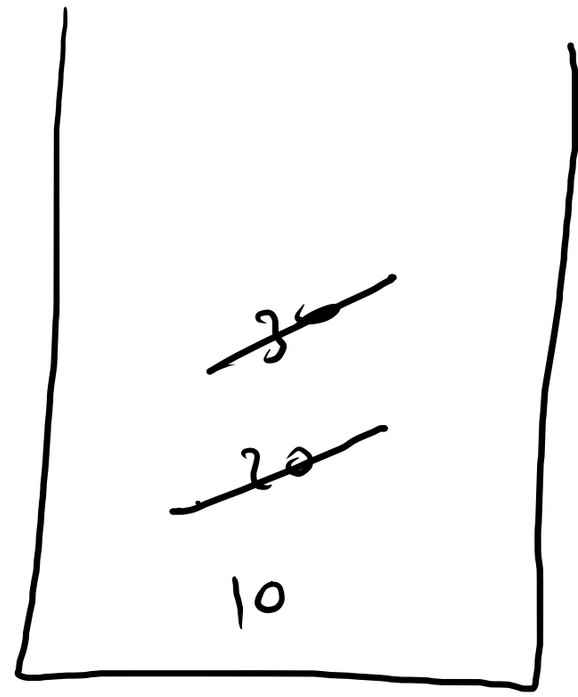


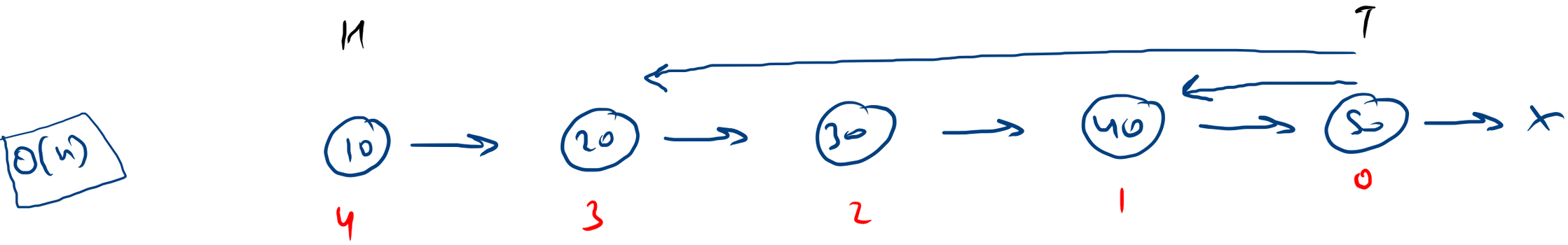
List



H

T





k= 1 → 40

k= 3 → 20

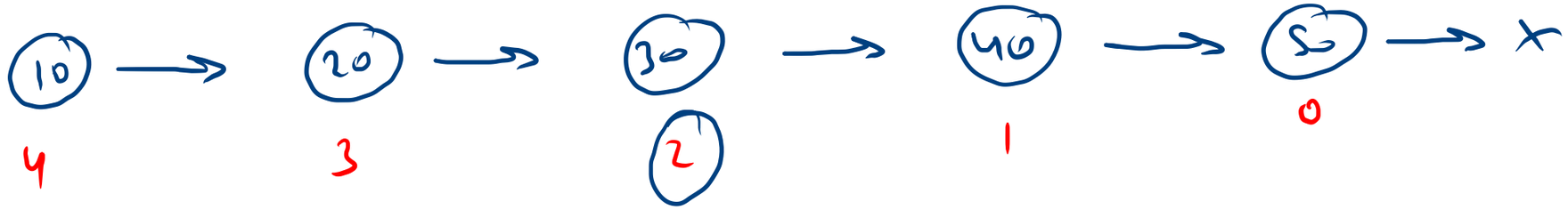
k= 4 → 10

size

head = 10

tail = 50

n



T

n1

n2

k= 2

O(n)

n x k

3 x k

2 x k

1 x k

0

n-1 O(n²)

n-2

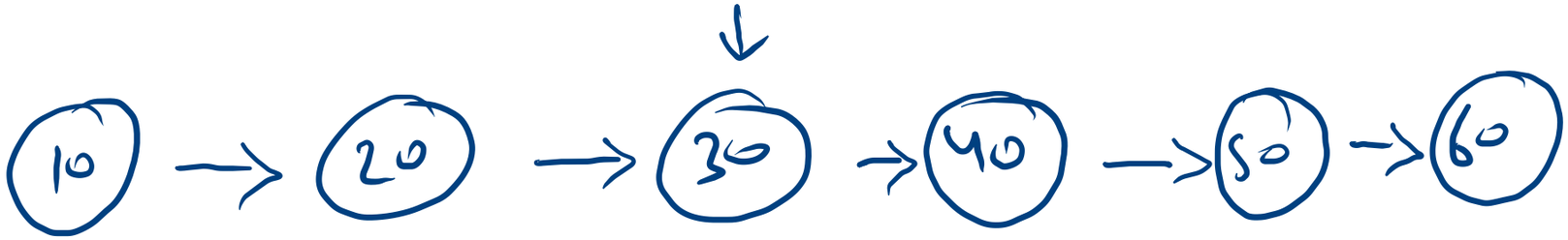
n-3

n-4

Iteration ✓
recursion ✓
size ✓

return 20

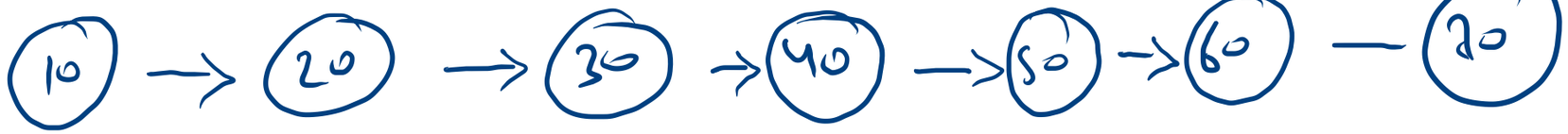
return 20



n1

n2

odd



slow
+1

fast
+2

even

slow

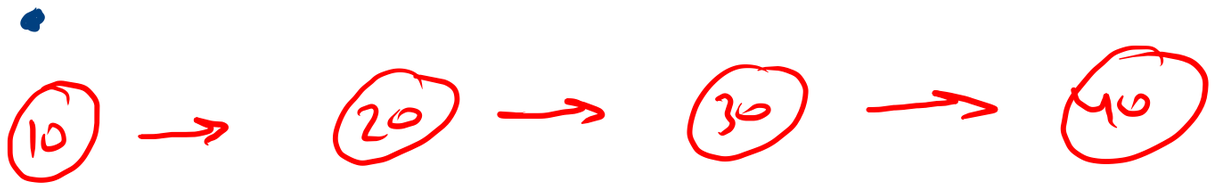
fast

slow = head

fast = head

while (fast != T &&
fast.next != T
slow = slow.next
fast = fast.next.next

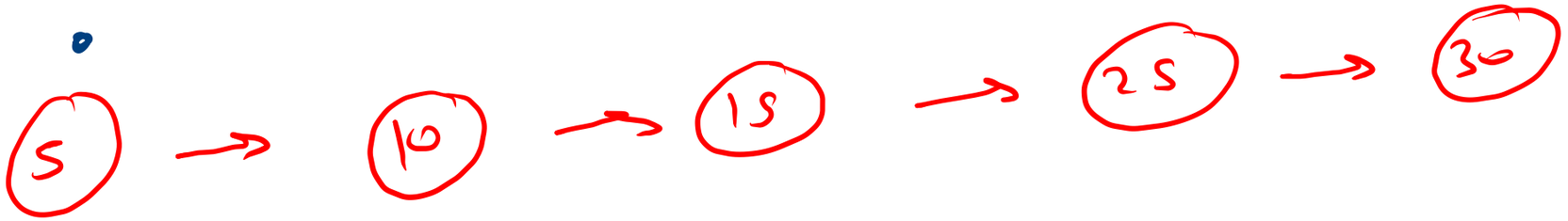
li



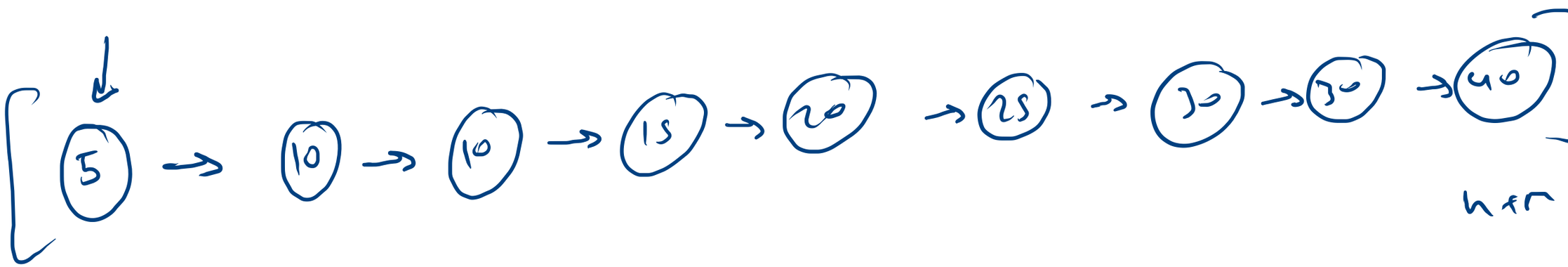
h

r

l2

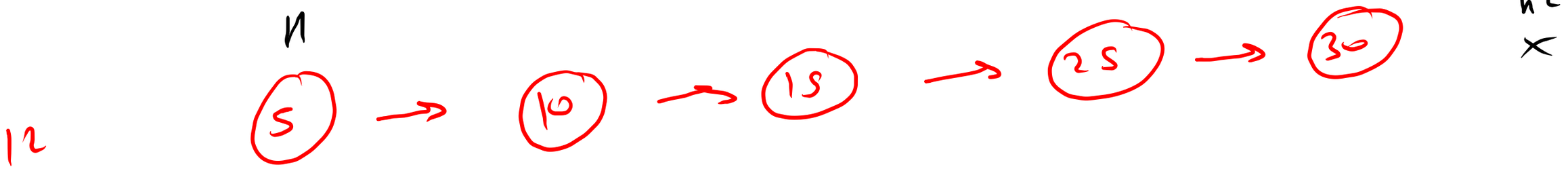


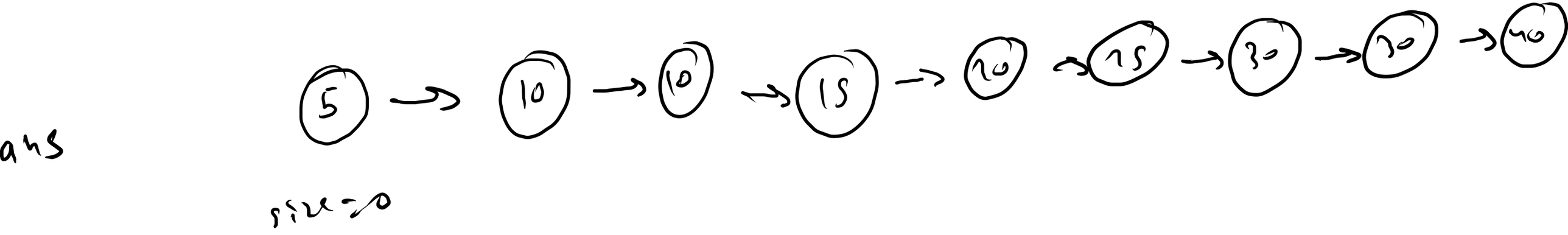
ans



h+r


l1
H
10
20
30
T
40
h1


l2
H
5
10
15
25
T
30
h2
X


ans
5
10
10
15
20
25
30
30
40