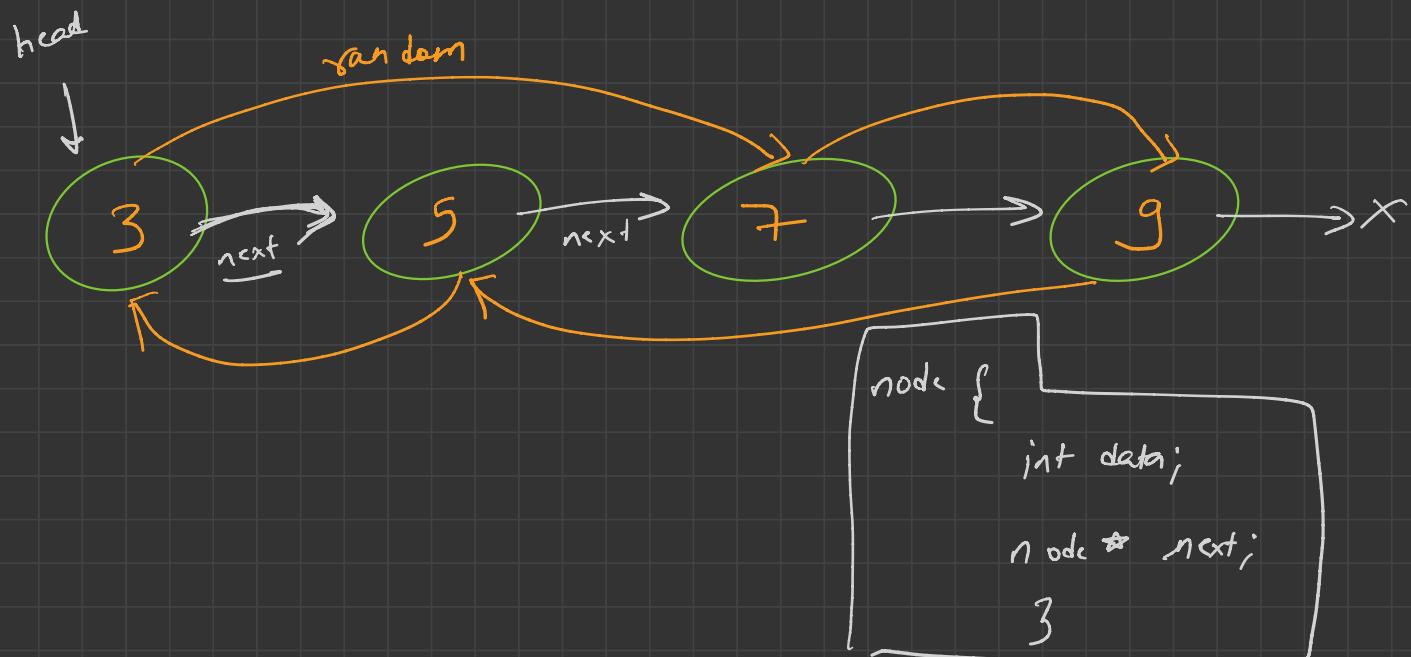
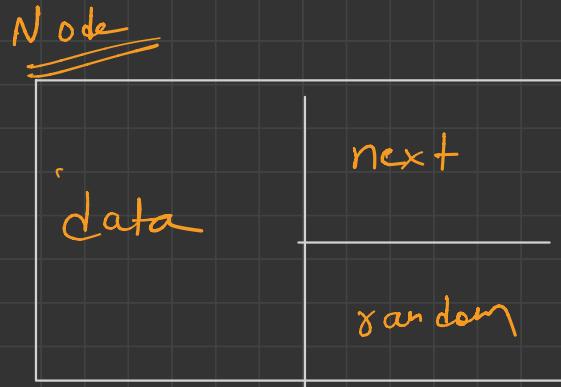



Clone a Linked List

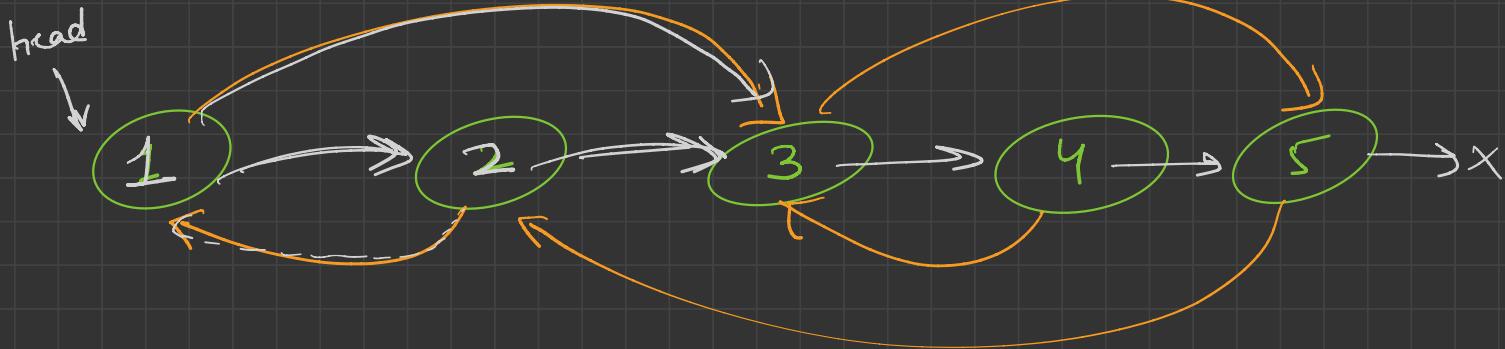


```
node  
{  
    int data;  
    node* next;  
    node* random;  
}
```



Approach:-

n/p



Intuition:

Approach #1

Create a clone List (using Original List next p(i))

O(n)

clone



↳ random ptr - copy -? $\rightarrow \underline{\underline{O(n^2)}}$

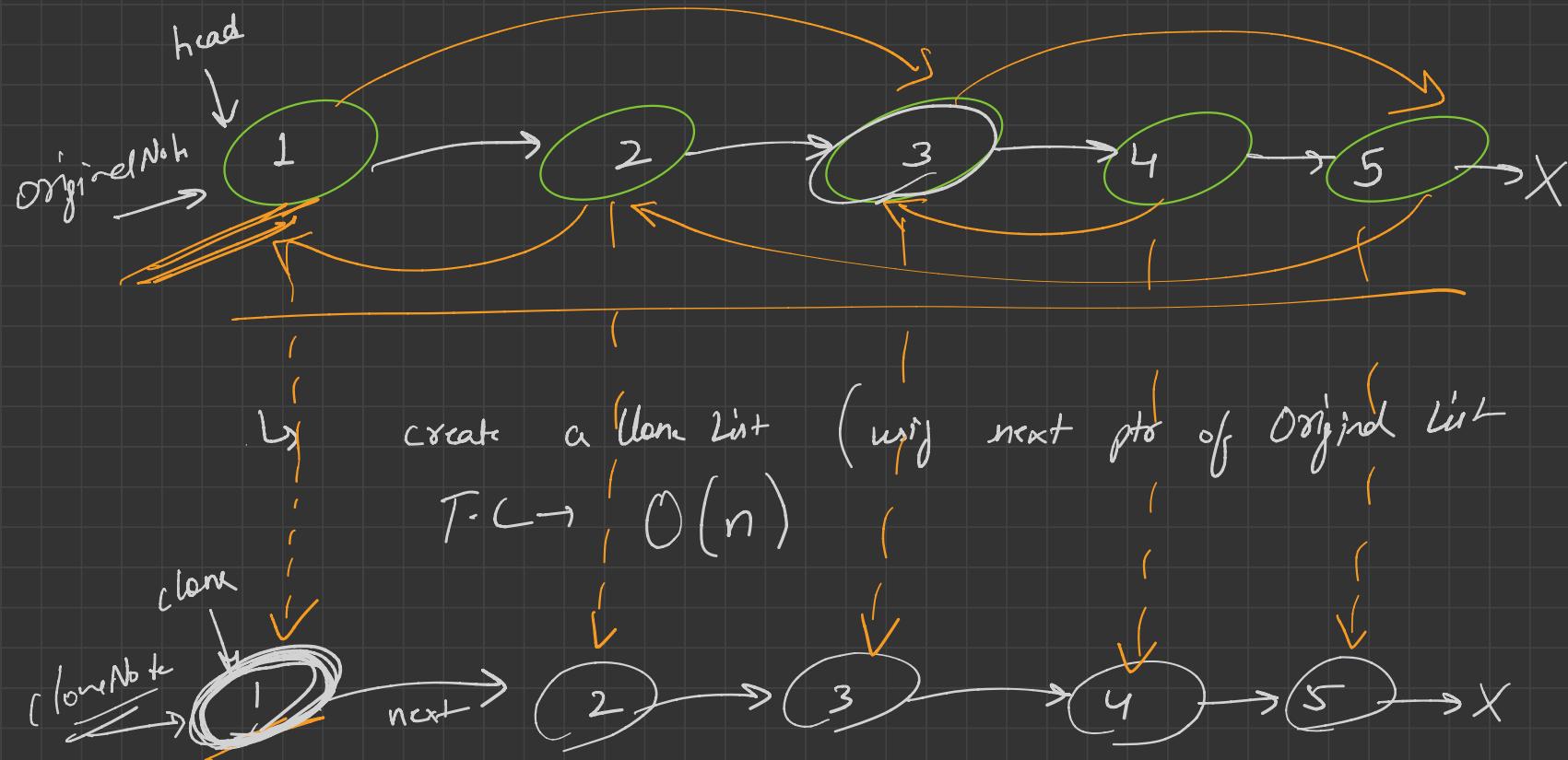
for ()

{ while ()

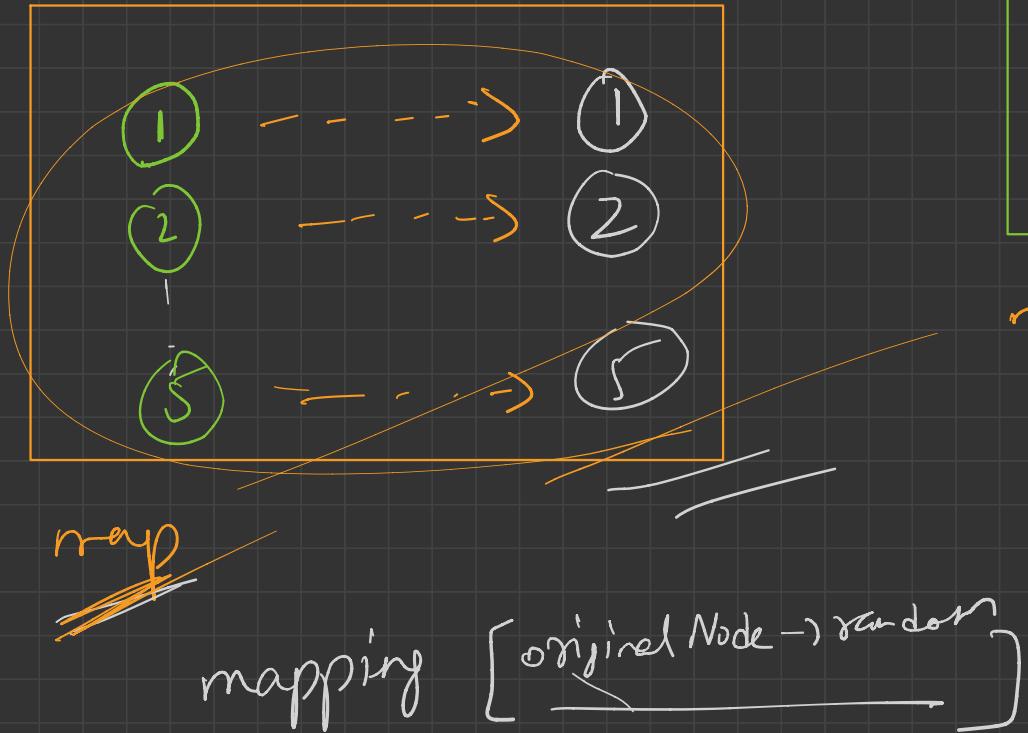
T.C $\rightarrow \underline{\underline{O(n^2)}}$

Approach #2

Original Node → Random



2 ↳ Random $p \neq 0 \rightarrow ?$



oldNode \rightarrow NewNode
OriginalNode ↲
cloneNode

mapping DS

map

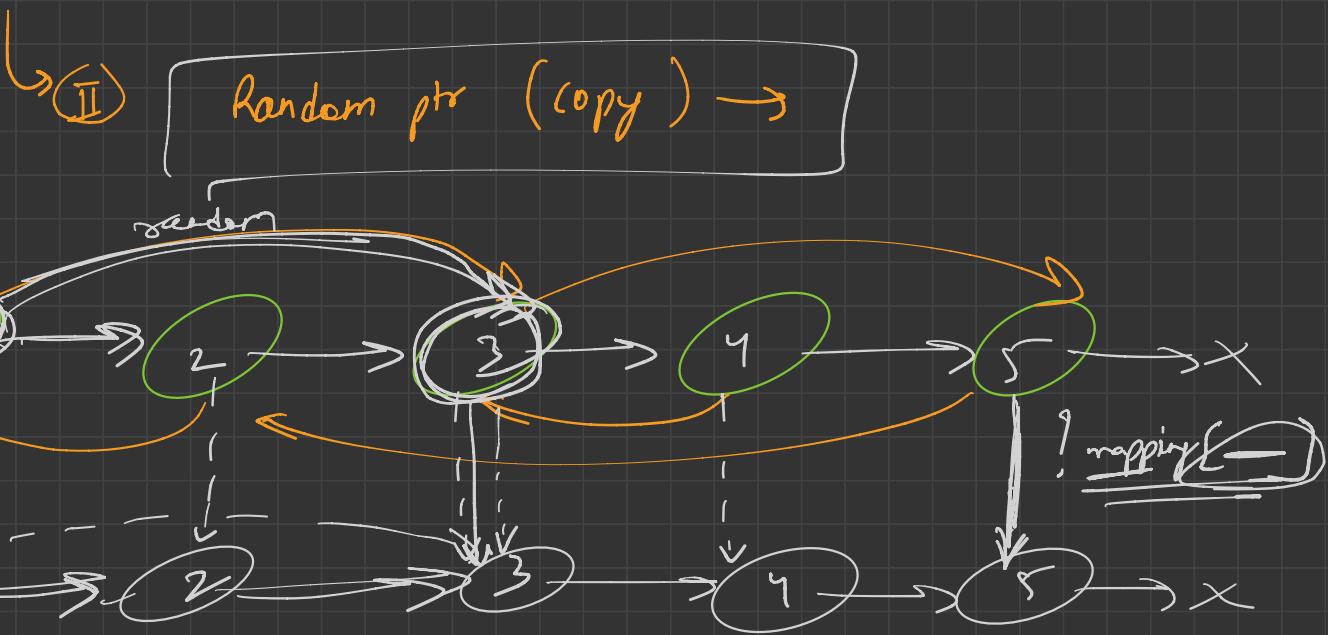
$\rightarrow \text{cloneNode} \rightarrow \text{random} = ?$

(2)

$\text{cloneNode} \rightarrow \text{random}$
= mapping [original Node \rightarrow Random]

Approach #2

→ (I) → create a clone List (using next ptr of Original List)



II

Clone Node \rightarrow random = ? = ?

mapping [Original Node \rightarrow random]

T.C \rightarrow $O(N)$

S.C \rightarrow $O(N)$

$\delta(1)$

map \propto

#3

$$T \cdot C \rightarrow O(N), \quad S \cdot C \rightarrow \underline{O(1)}$$

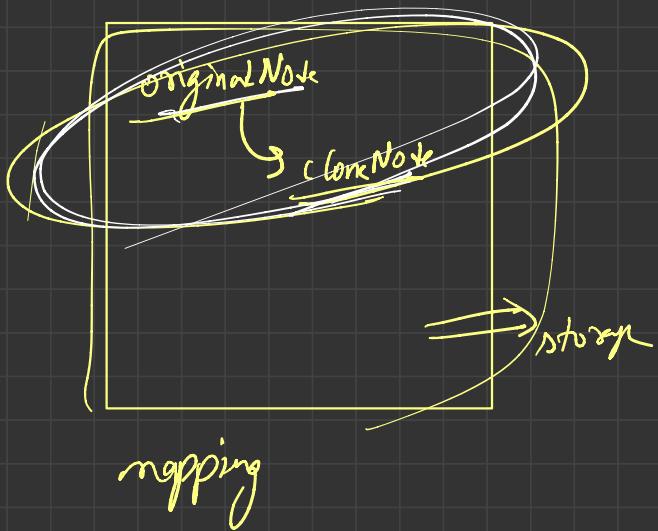
\Rightarrow changing links

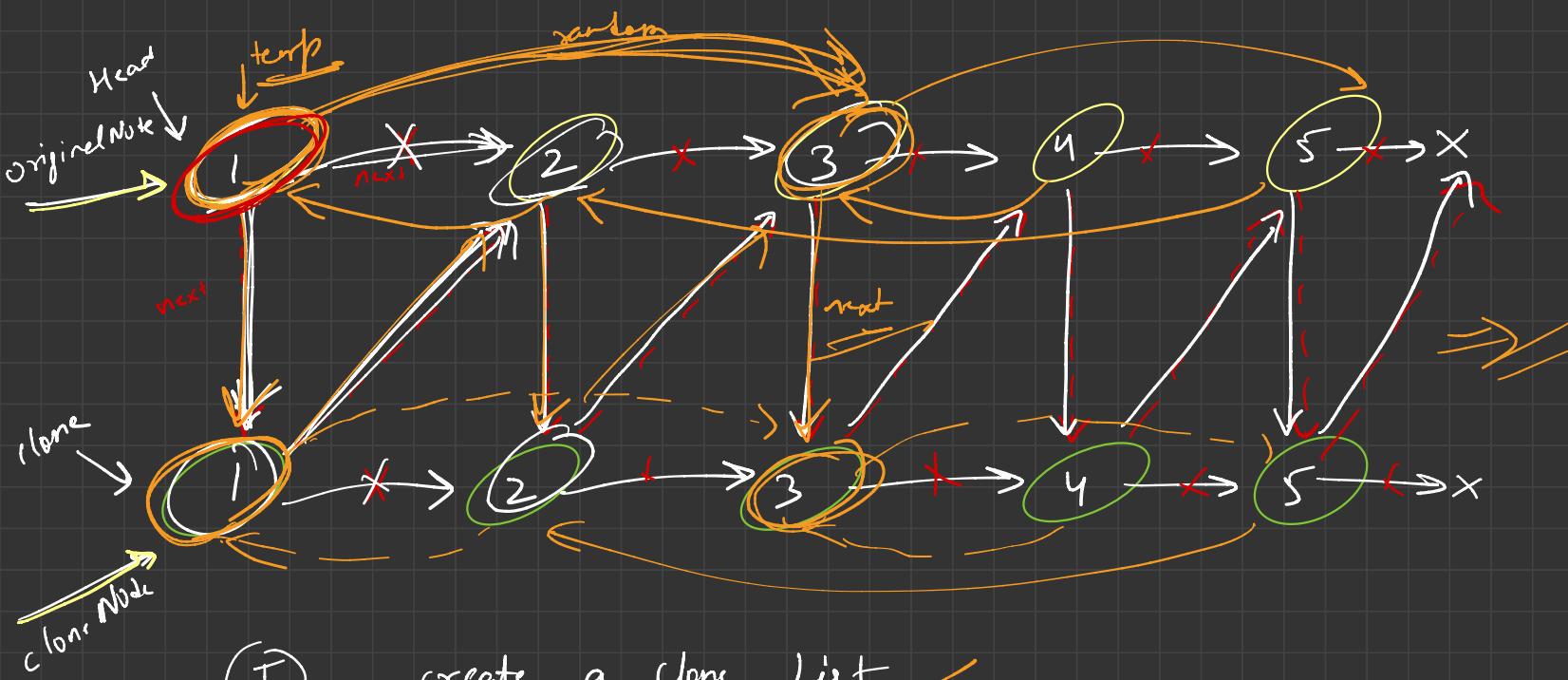
map \rightarrow we nhi Karna



Approach #3

$$O(N) \rightarrow O(1)$$





create a clone List

clone Nodes add in-between Original List

Random pointer [Clone List]

{

IV

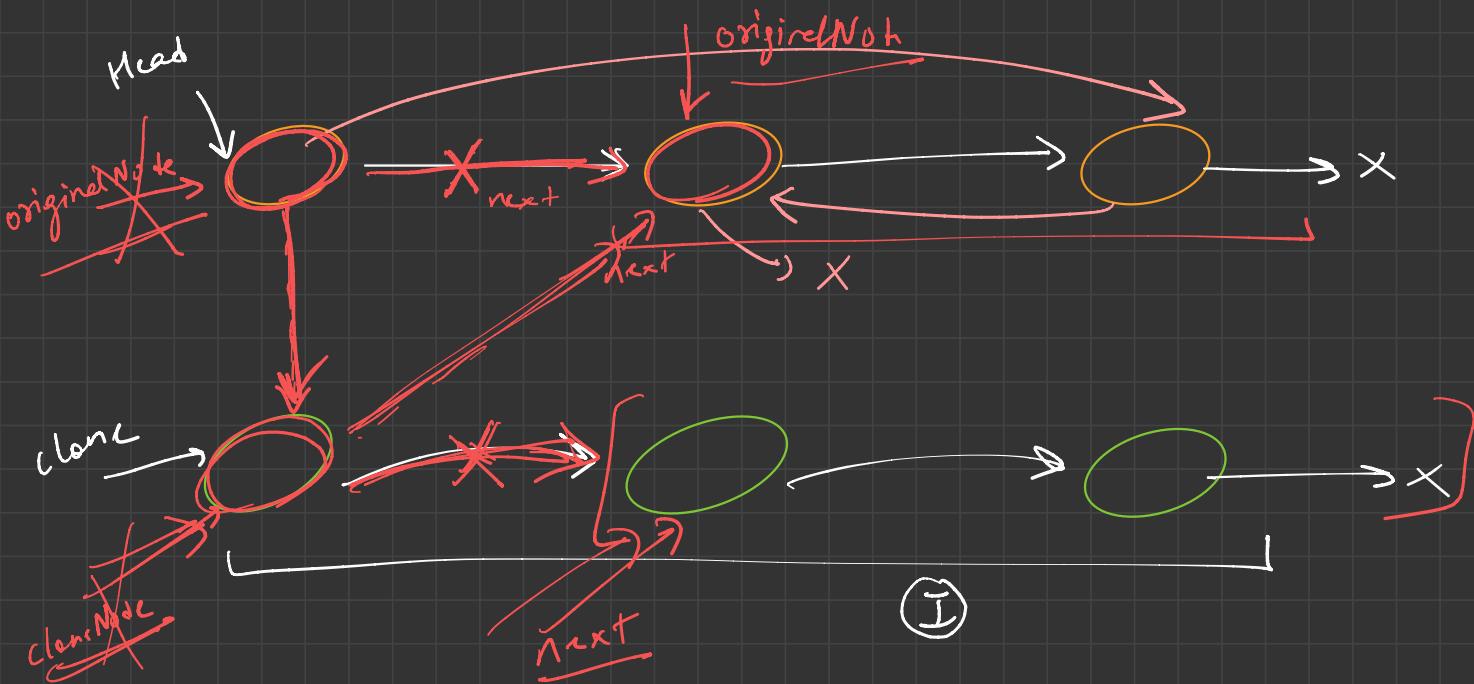
temp → next → random
= temp → random → next +

T.C → O(N)
S.C → O(1)

revert changes done in step 2

V

return ans {~~not~~ cloneHead }



$\left\{ \begin{aligned} \text{Node} * \text{next} &= \text{originalNode} \rightarrow \text{next}, \\ \text{originalNode} \rightarrow \text{next} &= \text{cloneNode} \\ \text{originalNode} &= \text{next} \end{aligned} \right.$

IV

.next = (cloneNode → next) //

(cloneNode → next) = originalNode

(cloneNode ← next)

-

original Node

