**Justin\_HashTable\_LinkedList\_0138.**  **Copy List with Random Pointer**

**Concept:**

用 recusion，把每個 node 分別接到所屬的位置

node.next 直接接到 head.next

node.random 直接接到 head.random

**Code:**

"""

# Definition for a Node.

class Node:

def \_\_init\_\_(self, x: int, next: 'Node' = None, random: 'Node' = None):

self.val = int(x)

self.next = next

self.random = random

"""

class Solution:

def \_\_init\_\_(self):

self.visitedHash = {}

def copyRandomList(self, head: 'Node') -> 'Node':

if head == None:

return None

if head in self.visitedHash:

return self.visitedHash[head]

node = Node(head.val, None, None)

self.visitedHash[head] = node

node.next = self.copyRandomList(head.next)

node.random = self.copyRandomList(head.random)

return node