138. Copy List with Random Pointer <medium>

class Solution {

unordered\_map<Node\*, Node\*> map;

public:

Node\* copyRandomList(Node\* head) {

if (!head)

return nullptr;

if (map.count(head))

return map[head];

Node\* newNode = new Node(head->val);

map[head] = newNode;

newNode->next = copyRandomList(head->next);

newNode->random = copyRandomList(head->random);

return newNode;

}

};

Node\* copyRandomList(Node\* head) {

if (head == nullptr)

return nullptr;

Node \* cur = head;

while (cur != nullptr)

{

cur->next = new Node(cur->val, cur->next, nullptr);

cur = cur->next->next;

}

cur = head;

while (cur != nullptr)

{

if (cur->random)

cur->next->random = cur->random->next;

cur = cur->next->next;

}

cur = head;

Node \* head\_cp = head->next;

Node \* cur\_cp = nullptr;

while (cur != nullptr)

{

cur\_cp = cur->next;

cur->next = cur->next->next;

cur = cur->next;

if (cur)

cur\_cp->next = cur->next;

}

return head\_cp;

}

};