

5月25日报

本日学习内容

1. 继续学习UI, UIWindow, UIViewController视图控制器的界面切换
2. 完成每日算法题
3. 写ui部分内容的博客

今日算法题

题目1: [138. 随机链表的复制](#)

138. 随机链表的复制

已解答 ✓

中等

🏷 相关标签

🏢 相关企业

💡 提示

Aa

给你一个长度为 n 的链表，每个节点包含一个额外增加的随机指针 `random`，该指针可以指向链表中的任何节点或空节点。

构造这个链表的 [深拷贝](#)。深拷贝应该正好由 n 个 **全新** 节点组成，其中每个新节点的值都设为其对应的原节点的值。新节点的 `next` 指针和 `random` 指针也都应指向复制链表中的新节点，并使原链表和复制链表中的这些指针能够表示相同的链表状态。**复制链表中的指针都不应指向原链表中的节点。**

例如，如果原链表中有 x 和 y 两个节点，其中 $x.random \rightarrow y$ 。那么在复制链表中对应的两个节点 x 和 y ，同样有 $x.random \rightarrow y$ 。

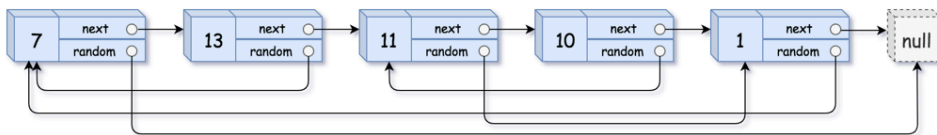
返回复制链表的头节点。

用一个由 n 个节点组成的链表来表示输入/输出中的链表。每个节点用一个 `[val, random_index]` 表示：

- `val`：一个表示 `Node.val` 的整数。
- `random_index`：随机指针指向的节点索引（范围从 `0` 到 `n-1`）；如果不指向任何节点，则为 `null`。

你的代码 **只** 接受原链表的头节点 `head` 作为传入参数。

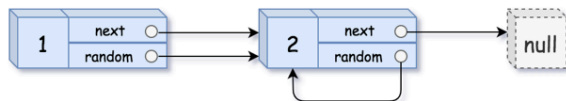
示例 1：



输入: `head = [[7,null],[13,0],[11,4],[10,2],[1,0]]`

输出: `[[7,null],[13,0],[11,4],[10,2],[1,0]]`

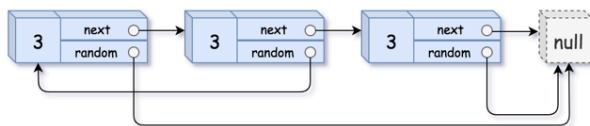
示例 2：



输入: `head = [[1,1],[2,1]]`

输出: `[[1,1],[2,1]]`

示例 3：



```
class Solution {
public:
    Node* copyRandomList(Node* head) {
        if (!head) return nullptr;

        unordered_map<Node*, Node*> oldToNew;

        for (Node* cur = head; cur; cur = cur->next) {
            oldToNew[cur] = new Node(cur->val);
        }
    }
};
```

```
for (Node* cur = head; cur; cur = cur->next) {
    oldToNew[cur]->next = oldToNew[cur->next];
    oldToNew[cur]->random = oldToNew[cur->random];
}

return oldToNew[head];
}
```

题目2: [48. 旋转图像](#)

48. 旋转图像

已解答

中等

🏷 相关标签

🏢 相关企业


Ax

给定一个 $n \times n$ 的二维矩阵 `matrix` 表示一个图像。请你将图像顺时针旋转 90 度。

你必须在 [原地](#) 旋转图像，这意味着你需要直接修改输入的二维矩阵。请不要使用另一个矩阵来旋转图像。

示例 1:

1	2	3
4	5	6
7	8	9

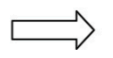


7	4	1
8	5	2
9	6	3

输入: matrix = [[1,2,3],[4,5,6],[7,8,9]]
输出: [[7,4,1],[8,5,2],[9,6,3]]

示例 2:

5	1	9	11
2	4	8	10
13	3	6	7
15	14	12	16



15	13	2	5
14	3	4	1
12	6	8	9
16	7	10	11

输入: matrix = [[5,1,9,11],[2,4,8,10],[13,3,6,7],[15,14,12,16]]
输出: [[15,13,2,5],[14,3,4,1],[12,6,8,9],[16,7,10,11]]

提示:

- `n == matrix.length == matrix[i].length`
- `1 <= n <= 20`
- `-1000 <= matrix[i][j] <= 1000`

```
class Solution {
public:
    void rotate(vector<vector<int>>& matrix) {
        int n = matrix.size();
        vector<vector<int>> tmp = matrix;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                matrix[j][n - 1 - i] = tmp[i][j];
            }
        }
    }
};
```

本日遇到的问题

1. UIViewController部分内容还是没有很好理解,

明日学习计划

1. 继续学习ui部分知识, 定时器与视图移动, Utextview和Utextfield