138. Copy List with Random Pointer

Description | Hints | Submissions | Discuss | Solution

Pick One

A linked list is given such that each node contains an additional random pointer which could point to any node in the list or null.

Return a deep copy of the list.

Seen this question in a real interview before? Yes No

Java

```java
/**
 * Definition for singly-linked list with a random pointer.
 * class RandomListNode {
 *     int label;
 *     RandomListNode next, random;
 *     RandomListNode(int x) { this.label = x; }
 * };
 */
```

```
    }
    /*
     * 如果要copy一个带有random pointer的list，主要的问题就是有可能这个random指向的位置还没有被copy到，所以解决方法都是多次扫描list。
第一种方法，就是使用HashMap来坐，HashMap的key存原始pointer，value存新的pointer。
第一遍，先不copy random的值，只copy数值建立好新的链表。并把新旧pointer存在HashMap中。
第二遍，遍历旧表，复制random的值，因为第一遍已经把链表复制好了并且也存在HashMap里了，所以只需从HashMap中，把当前旧的node.random作为key值，得到新
    public RandomListNode copyRandomList(RandomListNode head) {
            if(head == null) {
                return null;
            }

            HashMap<RandomListNode, RandomListNode> map = new HashMap<>();
            //newHead为新的链表。
            RandomListNode newHead = new RandomListNode(head.label);
            map.put(head, newHead);
            RandomListNode oldp = head.next;
            RandomListNode newp = newHead;

            while (oldp != null) {
             RandomListNode newnode = new RandomListNode(oldp.label);
             map.put(oldp, newnode);
             newp.next = newnode; //这个特别重要

             oldp = oldp.next;
             newp = newp.next;
        }

            oldp = head;
            newp = newHead;
            while (oldp!=null) {
             newp.random = map.get(oldp.random); //因为在第一步已经建立了旧到新的映射，所以get random就行
             oldp = oldp.next;
             newp = newp.next;
        }
            return newHead;
    }
}
```