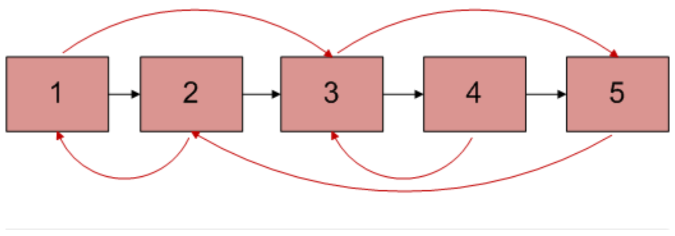
Clone a linked list with next and random pointer | Set 1

You are given a Double Link List with one pointer of each node pointing to the next node just like in a single link list. The second pointer however CAN point to any node in the list and not just the previous node. Now write a program in **O(n) time** to duplicate this list. That is, write a program which will create a copy of this list.

Let us call the second pointer as arbit pointer as it can point to any arbitrary node in the linked list.



Arbitrary pointers are shown in red and next pointers in black

Figure 1

Method 1 (Uses O(n) extra space)
This method stores the next and arbitrary mappings (of original list) in an array first, then modifies the original Linked List (to create copy), creates a copy. And finally restores the original list.

1) Create all nodes in copy linked list using next pointers.
3) Store the node and its next pointer mappings of original linked list.
3) Change next pointer of all nodes in original linked list to point to the corresponding node in copy linked list.
Following diagram shows status of both Linked Lists after above 3 steps. The red arrow shows arbit pointers and black arrow shows next pointers.

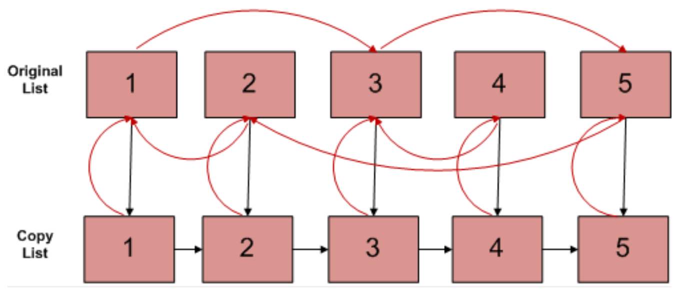


Figure 2

4) Change the arbit pointer of all nodes in copy linked list to point to corresponding node in original linked list.
5) Now construct the arbit pointer in copy linked list as below and restore the next pointer of nodes in the original linked list.

```
        copy_list_node->arbit =
                       copy_list_node->arbit->arbit->next;
        copy_list_node = copy_list_node->next;
```

6) Restore the next pointers in original linked list from the stored mappings(in step 2).

Time Complexity: O(n)
Auxiliary Space: O(n)

Method 2 (Uses Constant Extra Space)
Thanks to Saravanan Mani for providing this solution. This solution works using constant space.
1) Create the copy of node 1 and insert it between node 1 & node 2 in original Linked List, create the copy of 2 and insert it between 2 & 3.. Continue in this fashion, add the copy of N afte the Nth node
2) Now copy the arbitrary link in this fashion

```
     original->next->arbitrary = original->arbitrary->next;  /*TRAVERSE
TWO NODES*/
```

This works because original->next is nothing but copy of original and Original->arbitrary->next is nothing but copy of arbitrary.

3) Now restore the original and copy linked lists in this fashion in a single loop.

```
original->next = original->next->next;
copy->next = copy->next->next;
```

4) Make sure that last element of original->next is NULL.

Time Complexity: O(n)
Auxiliary Space: O(1)

Refer Following Post for Hashing based Implementation.
Clone a linked list with next and random pointer | Set 2