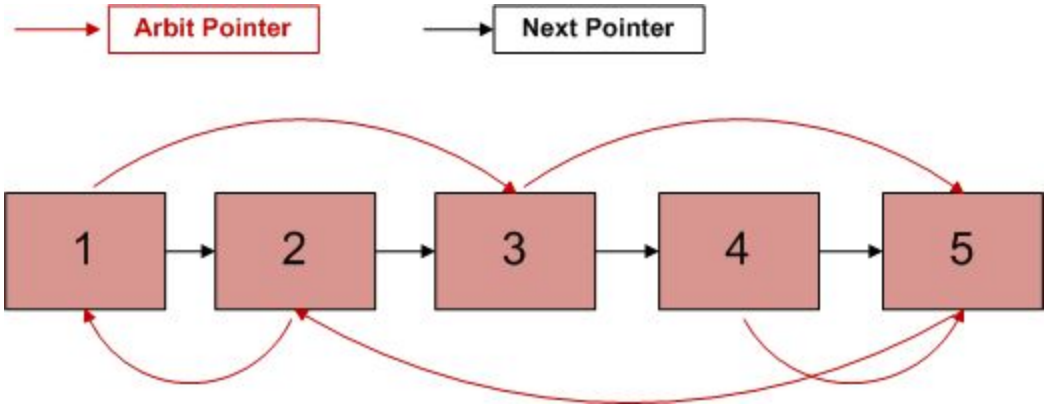

SOLUTION

We shall use the above double-linked list as an example to illustrate the solution. This original list is "List A". The algorithm shall create a duplicate list called "List B".

```
Node *duplicate_arbit_double_linked_list(Node *head) {

        While (iterate through List A) {
                Allocate Node for B
                Copy contents of current A node (A_i) to newly allocated node (B_i)
                B_i -> next = A_i -> next
                A_i -> next = B_i -> next
        } //allocate nodes for B, copy node content, and establish next pointers to create
        //a pathway that when followed will go: A1,B1,A2,B2,A3,B3,A4,B4,A5,B5

        While (iterate through List A) {
                B_i -> arbit = A_i -> arbit -> next
        } //exploit nature of structures and pointers to form arbit-pointer pathways for List B

        While (iterate through List A) {
                Node *bnode = A_i -> next; //note: this is B_i
                A_i -> next = A_i -> next -> next
                bnode -> next = bnode -> next ->next
        }

        Return pointer to head of List B
}
```

Note that each the three loops executes in time *n* complexity. Thus, the entire algorithm can be said to operate in O(n) time.