

3.26 Describe in a few sentences how to implement three stacks in one array.

Ans Three stacks can be implemented by having one grow from the bottom up, another from the top down and a third somewhere in the middle growing in some (arbitrary) direction. If the third stack collides with either of the other two, it needs to be moved. A reasonable strategy is to move it so that its center (at the time of the move) is halfway between the tops of the other two stacks.

3.29 Write an algorithm for reversing a singly linked list, using only constant extra space. This means that you cannot use recursion but you may assume that your algorithm is a list member function. Can such an algorithm be written if the routine is a constant member function?

Ans Reversal of a linked list can be done recursively using a stack, but this requires $O(N)$ extra space. The following solution is similar to strategies employed in garbage collection algorithms. At the top of the while loop, the list from the start to previousPos is already reversed, whereas the rest of the list,

```
void reverseList()
{
    Node currentPos, nextPos, previousPos;
    previousPos = null;
    currentPos = head.next; // skip header node
    nextPos = currentPos.next;
    while( nextPos != null)
    {
        currentPos.next = previousPos;
        previousPos = currentPos;
        currentPos = nextPos;
        nextPos = nextPos.next;
    }
    currentPos.next = previousPos;
    head.next = currentPos;
}
```

3.34 A linked list contains a cycle if, starting from some node n , following a sufficient number of next links brings us back to node n where n does not have to be the first node in the list. Assume that you are given a linked list that contains N nodes. However, the value of N is unknown.

Design an $O(N)$ algorithm to determine if the list contains a cycle. You may use only $O(1)$ extra space, that is, the amount of memory does not grow with N .

Ans Use two iterators p and q , both initially at the start of the list. Advance p one step at a time, and q two steps at a time. If q reaches the end there is no cycle; otherwise, p and q will eventually catch up to each other in the middle of the cycle.