How to create mergable stack?

Design a stack with following operations.

a) push(Stack s, x): Adds an item x to stack s

b) pop(Stack s): Removes the top item from stack s

c) merge(Stack s1, Stack s2): Merge contents of s2 into s1.

Time Complexity of all above operations should be O(1).

If we **use array** implementation of stack, then merge is not possible to do in O(1) time as we have to do following steps.
a) Delete old arrays
b) Create a new array for s1 with size equal to size of old array for s1 plus size of s2.
c) Copy old contents of s1 and s2 to new array for s1
The above operations take O(n) time.

We can **use a linked list** with two pointers, one pointer to first node (also used as top when elements are added and removed from beginning). The other pointer is needed for last node so that we can quickly link the linked list of s2 at the end of s1. Following are all operations.
a) push(): Adds the new item at the beginning of linked list using first pointer.
b) pop(): Removes an item from beginning using first pointer.
c) merge(): Links the first pointer second stack as next of last pointer of first list.

Can we do it if we are not allowed to use extra pointer?
We can do it with **circular linked list**. The idea is to keep track of last node in linked list. The next of last node indicates top of stack.
a) push(): Adds the new item as next of last node.
b) pop(): Removes next of last node.
c) merge(): Links the top (next of last) of second list to the top (next of last) of first list. And makes last of second list as last of whole list.

This article is contributed by **Rahul Gupta**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above