**MergeableStack**

**Q:** 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.

**HINT:** Use linked list as it will be easy for merging.

**Algorithm:**

* Maintain 2 pointers for first stack, one for head and one for its end.
* Maintain 1 pointer for the head of the second stack.
* When merge is called, end pointer of first stack will point to the head of second stack.

**Example:**

* Stack1: 1->2->3->4
* Stack2: 5->6->7->8
* merge(): 1->2->3->4->5->6->7->8

**Complexity:**

Time Complexity: O(n+m) (while loop for printing)

where, n are the elements in first stack and,

m are the elements in the second stack.