**NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES**

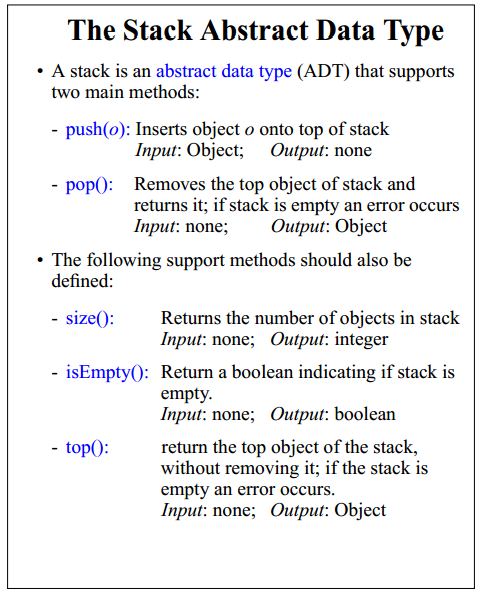
**CL 201-Data Structures**

**Lab Session 09**

**STACK**

A stack is a container of objects that are inserted and removed according to the last-in-ﬁrst-out (LIFO)

Principle. Objects can be inserted at any time, but only the last (the most-recently inserted) object can be removed. Inserting an item is known as “pushing” onto the stack. “Popping” off the stack is synonymous with removing an item.

****

**Task1: Create a stack, Implement all functions**

**Merge Sort:**

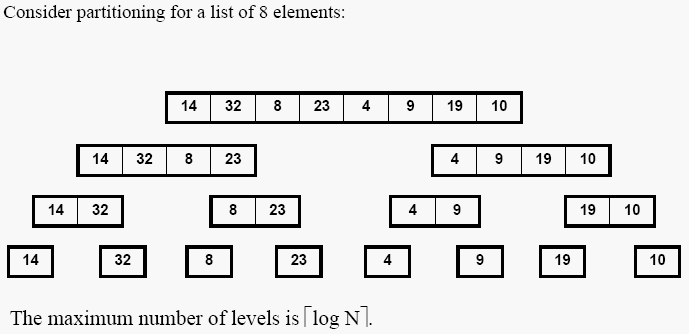
Merge Sort which is able to rearrange elements of a list in ascending order. Merge sort works as a

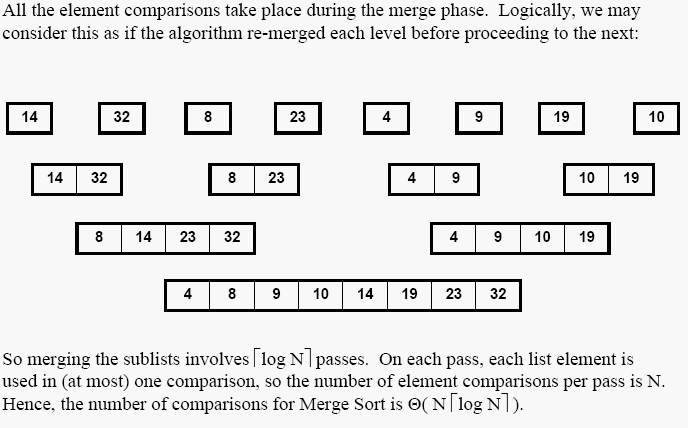
Divide-and-conquer algorithm. It recursively divide the list into two halves until one element left, and merge the already sorted two halves into a sorted one.

**APPROACH**

Merge sort uses a divide-and-conquer approach:

1. Divide the array repeatedly into two halves
2. Stop dividing when there is single element left. By fact, single element is already sorted.
3. Merges two already sorted sub arrays into one.





Pseudo Code:

* Input: Array A[1…N], indices p, q, r (p ≤ q <r).
* A[p…r] is the array to be divided
* A[p] is the beginning element and A[r] is the ending element
* Output: Array A[p…r] in ascending order

**MERGE-SORT(A,p,q,r)**

**1 if p <r**

**2 then q ← (r + p)/2**

**3 MERGE-SORT(A, p, q )**

**4 MERGE-SORT(A,q+1,r)**

**5 MERGE(A, p, q, r)**

**Task2: Implement the algorithm, and find total number of calls to MERG\_SORT.**

**QUICK SORT**

Quick sort is a very efficient sorting algorithm invented by C.A.R. Hoare. It has two phases:

* The partition phase and
* The sort phase.

**HOME TASK: Identify the total number of comparisons in Quick sort, for sorted array(best case), unsorted array (reverse order (worst case) and average case.**