Assignment - 40
DSA: Stack and Queue

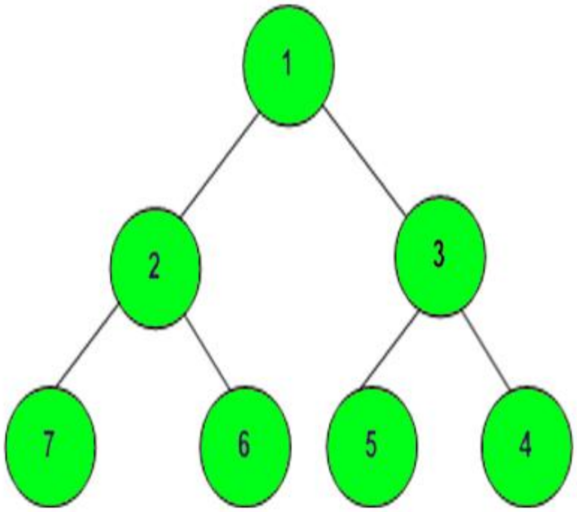
1. Given a stack with push(), pop(), empty() operations, delete the middle of the stack without using any additional data structure.
   Middle: ceil((size_of_stack+1)/2) (1-based index)
   Example 1:

   Input:
   Stack = {1, 2, 3, 4, 5}
   Output:
   ModifiedStack = {1, 2, 4, 5}
   Explanation:
   As the number of elements is 5, hence the middle element will be the 3rd element which is deleted

   Example 2:
   Input:
   Stack = {1 2 3 4}
   Output:
   ModifiedStack = {1 3 4}
   Explanation:
   As the number of elements is 4, hence the middle element will be the 2nd element which is deleted

2. Given an expression string x. Examine whether the pairs and the orders of “{“,”}”,”(“,”)”,”[“,”]” are correct in exp.
   For example, the function should return 'true' for exp = “[()]{}{[()()]()}” and 'false' for exp = “[(])”.

3. Complete the function to find spiral order traversal of a tree. For the tree below, the function should return 1, 2, 3, 4, 5, 6, 7.



Example 1:

```
Input:
     1
   /     \
  3       2
Output:1 3 2
```

Example 2:

```
Input:
                 10
              /      \
            20     30
           /  \
       40   60
Output: 10 20 30 60 40
```

4. Given a stack, the task is to sort it such that the top of the stack has the greatest element.
   Example 1:
   Input:
   Stack: 3 2 1
   Output: 3 2 1
   Example 2:
   Input:
   Stack: 11 2 32 3 41
   Output: 41 32 11 3 2

5. Reverse the string using stack.

6. Create a Queue using Array or Dynamic Array.
7. Create a Queue using a linked list.
8. Efficiently implement k Queues in a single array.
9. Implement Priority Queue using Linked Lists.
10. Given Nodes with their priority, implement a priority queue using a doubly linked list.