

CIRCULAR LINKED LIST

Name- Harsh Mishra

Roll number- 19049

Major changes need to be done:

A circular linked list is a type of linked list in which the first and the last nodes are also connected to each other to form a circle. The following major changes need to be done when creating a circular linked list-

1. Creating a link between first and last node by storing the address of the first node in the last node.
2. In every loop when the last node is created, a link is established between the last node and the first.
3. If list is Empty then, set head = newNode and newNode→next = head .
4. If it is not Empty then, define a Node pointer 'current' and initialize with 'head'.
5. Keep moving the 'current' to its next node until it reaches to the last node (until 'current → next == head'). Finally, set 'newNode → next = head', 'head = newNode' and 'current → next = head'.
6. The same is taken care of while reversing. Head is going to be our last node after reversing the list and we iterate till the initial node in the circular list is reached.

Advantages of Circular Linked List:

- 1) Any node can be a starting point. Entire list can be traversed from any node.
- 2) It saves time when we have to go to the first node from the last node(time complexity $O(1)$). It can be done in single step because there is no need to traverse the in between nodes. But in double linked list, we will have to go through in between nodes.
- 3) Circular lists are useful in applications to repeatedly go around the list. For e.g., it is convenient for the operating system to use a circular list so that when it reaches the end of the list(of running applications) it can cycle around to the front of the list.
- 4) It is useful for implementation of queue and also advanced data structures like Fibonacci Heap.

Disadvantages of Circular Linked List:

1. Reversing of circular list is complex as compared to singly or doubly linked lists.
2. Since the last node in a circular linked list refers to the starting node, if not traversed carefully, then you could end up in an infinite loop.
3. If we at a node and go back to the previous node, then we can't do it in single step instead we have to complete the entire circle by going through the in between nodes and then we will reach the required node.
4. Like singly and doubly lists, circular linked lists also do not support direct accessing of elements.