In a singly circular linked list, each node has one link, except that the next link of the last node points back to first node. In circular linked lists, each node has a successor.

**Advantages -**

* When several processes are using the same computer resource for the same amount of time, we have to assure that no process accesses the resource before all other processes do (**round robin algorithm)**.
* Useful for implementation of queue. A pointer can be maintained to the last inserted node and front can always be obtained as next of last.
* Fibonacci Heap

**Split a circular linked list into two parts**

If number of nodes in the list are odd then make first list one node extra than second list

*Approach-*

* Store the mid and last pointers of the circular linked list using Floyd cycle finding algorithm
* Make the second half circular
* Make the first half circular
* Set head pointers of the two linked lists

Time Complexity: O(n)

Space Complexity: O(1)