

ADITYA RAJ
U20CS100
DS TUTORIAL - 9

Algorithm

- 1) Linked List is empty:
 - a. since new_node is the only node in CLL, make a self loop.
- 2) new_node->next = new_node;
 - a. change the head pointer to point to new node.
- 3) *head_ref = new_node;
- 4) New node is to be inserted just before the head node:
 - i. Find out the last node using a loop.
- 5) while(current->next != *head_ref)
 - a. current = current->next;
 - i. Change the next of last node.
- 6) current->next = new_node;
 - i. Change next of new node to point to head.
- 7) new_node->next = *head_ref;
 - i. change the head pointer to point to new node.
- 8) *head_ref = new_node;
- 9) New node is to be inserted somewhere after the head:
 - i. Locate the node after which new node is to be inserted.
- 10) while (current->next!= *head_ref &&
 - a. current->next->data data)
- 11) { current = current->next; }
 - i. Make next of new_node as next of the located pointer
- 12) new_node->next = current->next;
 - i. Change the next of the located pointer
- 13) current->next = new_node;

Binary search for finding element and delete it:

Algorithm

1. If the list is empty, return.
2. If the list is not empty, create two nodes, say, current and previous.
Current will point to the head and previous will point to NULL.

3. Traverse using the current pointer to find the node that is to be deleted. In every traversal, we have to make the current as previous as this is a doubly circular linked list.
4. Now, if the node is found but is the only node present in the list, then we will make the head NULL and return.
5. If the found node is the head node i.e. if (`current==head`), move previous to the last node. Now, do `head = head -> next`. The node has been deleted successfully, but now, the last node will point to the new head and to do so make `prev -> next = head` and `head -> prev = previous`. By doing this, all the required links have been changed. In the end, free the node pointed by the current.
6. If the found node is the tail node i.e. if (`current -> next == head`), then the second last node will now point to the head node and to do so make `previous -> next = head` and `head -> prev = previous`. By doing this, the links have been successfully changed. In the end, free the node pointed by the current.
7. If the found node is neither the first nor the last, then we will simply store the next of current in temp. Now, the previous will point to the temp and the prev of temp will point to the previous. In the end, free the node pointed by the current.