23-09-2020

AKSHAY MITTUR
1BM18CS010

ADS LAB
WEEK 1 WRITE UP

⇒ Implement Memory efficient doubly linked list

⇒ using XOR

Pseudo Code :

Insert at beginning

```
func insertbeg (Node n, int data) :{
    newnode = new Node();
    newnode → data = data;
    newnode → npx = n;
    if( n) n → npx = XOR (newnod
    n = newnode;
}
```

Insert at End

```
func insertEnd (Node n, int data) {
    Node newn = new Node();
    newn → data = data;
    if (!n) { newn → npx = n;
              n = newn;
            }
    else{   Node curr = n;
            Node prev = NULL;
            Node next;
    while ( XOR( prev, curr → npx ) != NULL){
            next = XOR( prev, curr → npx );
            prev = curr;
            curr = next;
    }
```

23-09-20

AKSHAY MITTUR
1BM18CS010

```
        newn -> npx = curr;
        curr -> npx = XOR(prev, newn);
    }
}
```

XOR function:

```
func XOR(Node a, Node b) {
    return (Node)(uintptr_t)(a) ^ (uintptr_t)(b);
}
```