Lecture - 33

Linked List Advance

Page No.:
Date: / /

* **Code : (Remove loop in Linked List)**

O(N)

```
if ( head == head → next)
    head → next = NULL;

if ( ! head → next)
    return

Node * slow = head → next;
Node * fast = head → next → next;

while ( Fast && Fast → next && Fast != slow ) {
        Fast = Fast → next → next;
        slow = slow → next;
};

if ( !Fast || !Fast → next )
    return

Fast = head;

if ( fast == slow ) {
    while( slow → next != Fast )
        slow = slow → next;
    slow → next = NULL;
    return ;
}

while( Fast → next != slow → next ) {
    fast = fast → next;
    slow = slow → next;
}
```

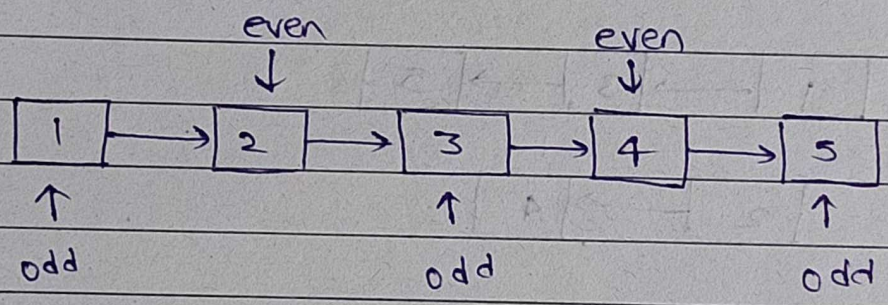
Teacher's Signature.....

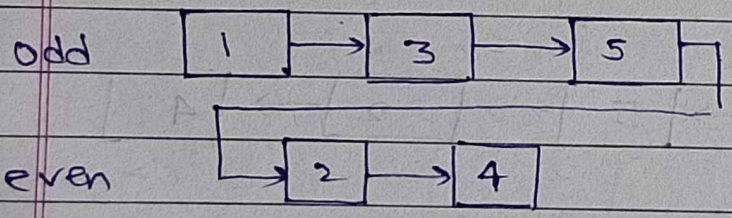
```
slow → next = NULL;
return;
}
```

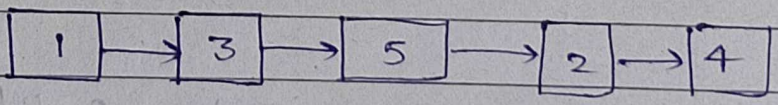
* **Rearange Linkedlist :**

Given : A single linked list

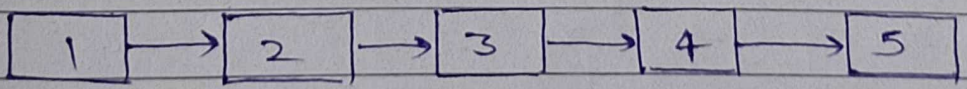
Rearrange like : odd position Node Comes first then Even position Node Comes



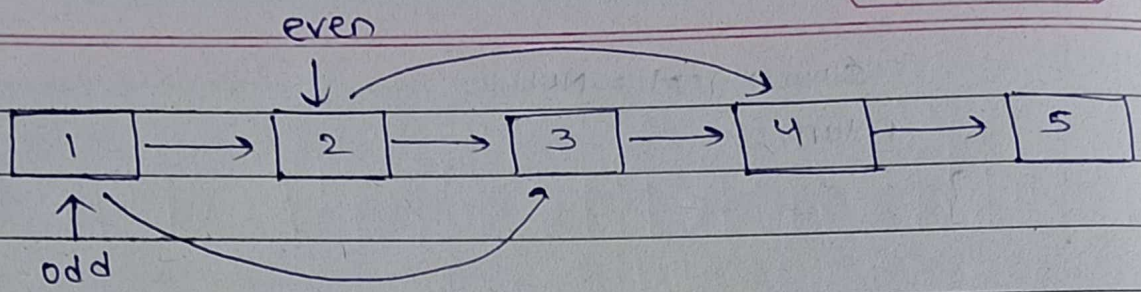


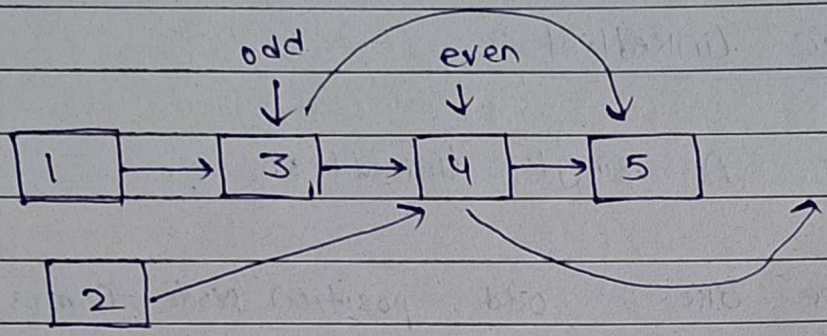
Final Answer: 

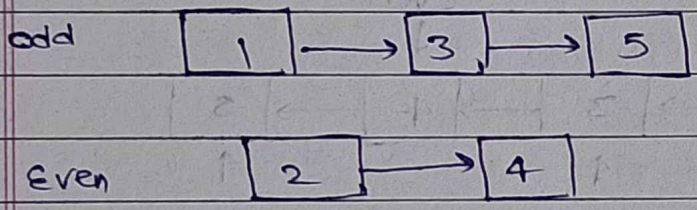
How to solve this logically.



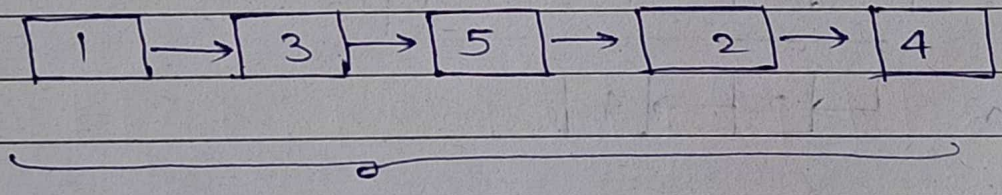
Teacher's Signature.....







→ Join odd & even



output.

Time Complexity : O(N)
Space Complexity : O(1)

Code :

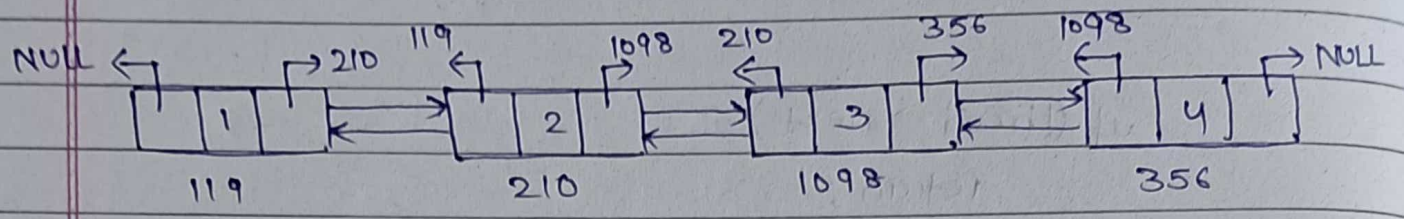
```
if (!head -> next)
    return ;

Node * first = head;
Node * second = head -> next;
Node * temp = head -> next;

While ( second && second -> next ) {
        first -> next = second -> next;
        first = first -> next;
        second -> next = first -> next;
        second = second -> next;
    };

first -> next = temp;
```

* Till Now, We can only access the Node which is present forwant.

It we want to access the Node which is present backward, we have to use circular linked list. But it takes O(N) time Complexity.

If we want to access in O(1) time Complexity then we use the doubly linked list.

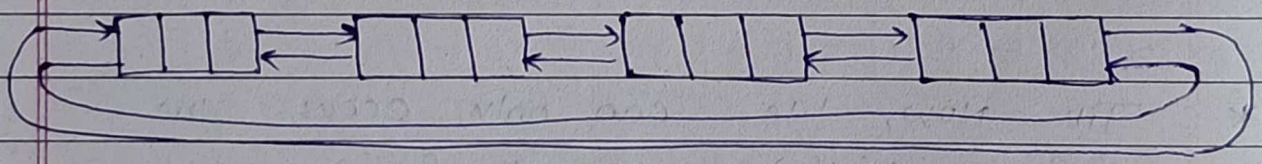
Teacher's Signature.....



on this we can access previous element in O(1) time.

```
class Node {
    public:
     int data;
     Node* next;
     Node* prev;
    }
```

* Circularly Doubly linked list :



* Convert the array int doubly linked list

arr = [1, 2, 3, 4, 5]

Teacher's Signature.....

```
class Node {
    public:
    int data;
    Node * prev;
     Node* next;
};

Node (int x) {
    data = x;
    prev = NULL;
    next = NULL;
  }
}

int main () {

    int arr [5] = {1, 2, 3, 4, 5};
     Node * head;
    head = new Node (arr [0]);
     Node * first = head;

  for(int i = 1;  i<5;  i++) {
    first → next = new Node (arr [i]);
       first → next → prev = first;
          first = first → next;
   };
   first = head;
     while ( first) {
          cout << first → data << " ";
            first = first → next;
       }
    return 0;
}
```

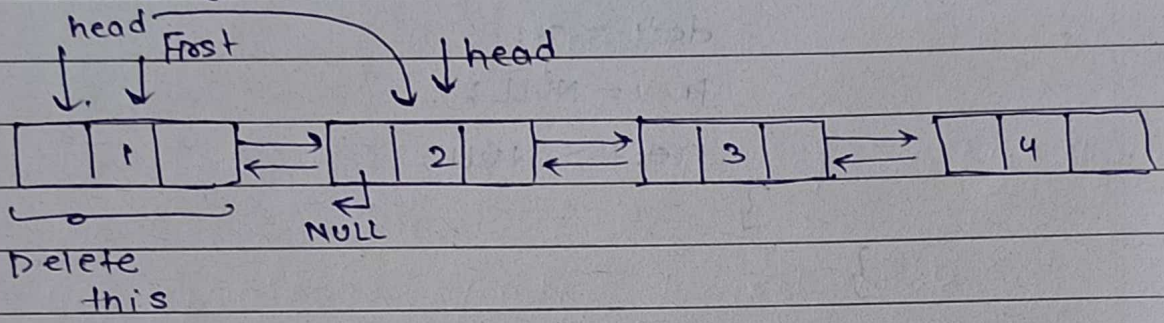
Teacher's Signature.....

Page No.:
Date: / /

* Deletion in doubly linked list :

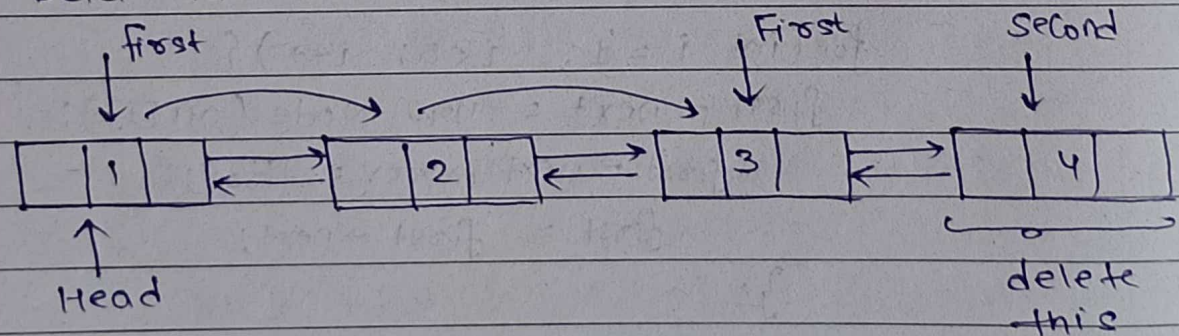
1. Delete first Node
2. Delete last Node
3. Delete Mid Node

* Delete first Node :




```
Node* first = head;
head = head->next;
head->prev = NULL;
Delete (first);
```

* Delete last Node :




```
Node * first = Head;
While (First -> next) {
    first = first -> next;
}
Node * second = first -> next;
```

Teacher's Signature.....