

CHECK_IF_A_LINKED_LIST_IS_A_PALINDROME

★ In this problem, we are given a Linked List and our job is to return true if the linked list is a palindrome otherwise false.

Brute force solution is to use a data structure like a stack or array to store all elements and then iterate back to check for palindrome. Time Complexity is $O(2N)$ and Space Complexity is $O(N)$.

Optimal Solution involves 4 steps
Finding the Middle
Reversing the Second Half
Comparing for Palindrome
Fix the second half

C++ Code :

```
bool palindromeLL(Node* head) {  
    Node* s = head;  
    Node* f = head;  
    while (f->next != nullptr && f->next->next  
           == nullptr) {  
        s = s->next;  
        f = f->next->next;  
    }  
    Node* nh = reverseLL(s->next);
```

f = head ;

Node * sec = nH ;

while(sec != nullptr) {
 if(sec->data != f->data) {

reverse(nH) ;

return false ;

}

f = f->next ;

sec = sec->next ;

}

reverse(nH) ;

return true ;

}