Linear Data Structures Continued.... [Classes]

Stack → [LIFO] (Last In First Out) (Stack) java.util

Applications :→ Reversal Operation

example :→ (I) Palindrome Check

(II) Queue Reversal

-1 empty, 0

top = -1+1 = 0 → 1 → 2 → 2 → 1 → null

top++

↳ Palindrome Linked List

#include <stack>

Queue → | 1 | 2 | 3 | 4 |

reverse → | 4 | 3 | 2 | 1 | stack

size ( ) → (top+1)

| x | y | z | w | → (w)

0 1 2 3 top

top = -1

-1+1 = 0

size = 0

push ( ), pop ( ), size ( ), empty ( ), full ( )

Strings (1) → Create a generic stack which can accept any data.

100

false

'c'

(6-8 LPA)

A/C

⇓

Infosys

Annotations ?

@override

Generics → <T>

Wildcards <?>

(Object)

(Linked List + Stack) Linked List Palindrome

Stack< >

(SLL) temp 1 → 2 → 2 → 1 → null

head (temp) stop top

T-1 temp

T-2

1 2 2 1 same true

1 — 2 — 3 1 false

1 3 2 1 false

Stack

Leet Code Questions on Linked Lists :

(I) Middle node of a linked list 1—2—3—4—5

(II) Delete middle node of linked list 1—2—3—4—5—6 ↳ 4

[Floyd's Algorithm :→] (Slow & fast pointer)

The Hare & Tortoise Algorithm

(III) Cycle detection in a linked list 1—2—3—4

(IV) 1, 2, 3 | 4, 5, 6 → Sum + carry

LL1 LL2

5 → 7 → 9 → null

6 LPA — 22 LPA — 46 LPA

3 mins

1823

(Josephus Problem)

Leet Code

Coding Ninjas

slow slow fast slow

1 — 2 — 3 — 4

fast fast slow fast

Detect A Cycle in Linked List

(Floyd's Algo)

Hare & Tortoise

slow = head

2 fast = head

```
while (fast && fast.next != null)
    slow = slow.next;
    fast = fast.next.next;
    if (slow = fast);
        return true;
}
return false.
```

0 1

f

1 — 2 — 2 — 4 — 5 null

s s f s f

n = 5|2 = 2