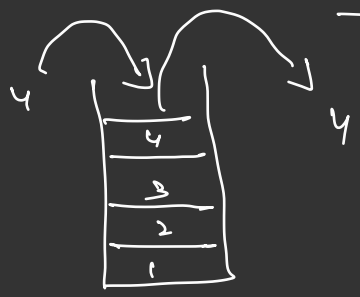
STACK

↳ Data Structure

→ Linear DS

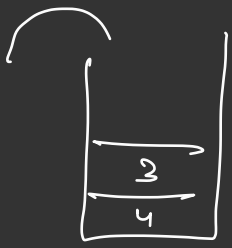
→ Last In First Out (LIFO)



eg → Call / Function Stack in memory

→ Undo / redo

→



Functions of stack

O(1)

1. Push → add at top
2. Pop → remove from top
3. Peek → return top
4. Size → returns size of stack
5. IsEmpty → returns boolean depending on size

push(4)
push(3)
push(5)
pop()

Stack {

size

}

1. Push → addFirst
2. Pop → removeFirst
3. Peek → getFirst → getNode(0)
4. size
5. IsEmpty → size == 0


```
push(val) {
    arr[++top] = val;
}

pop() {
    top--;
    return arr[top+1];
}

peek() {
    return arr[top];
}
```

Ques Duplicate bracket

((a + b) + c) // false

→ (((a+b)) + c) // true

(((a + b)) + c)

i

Ques Balanced brackets

Balanced {a + [b + (c+d)]} ← {[](())}

Not balanced

{a + [b + (c+d)]}

{a + [b + (c+d)]}

{a +]}

{[(]}

{{