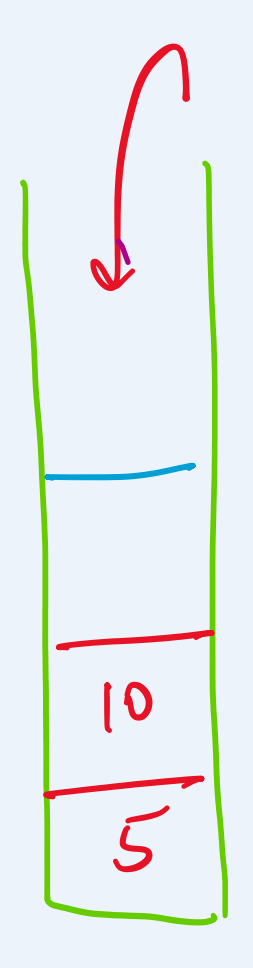
Friday, 26 March 2021 8:34 PM

Stack

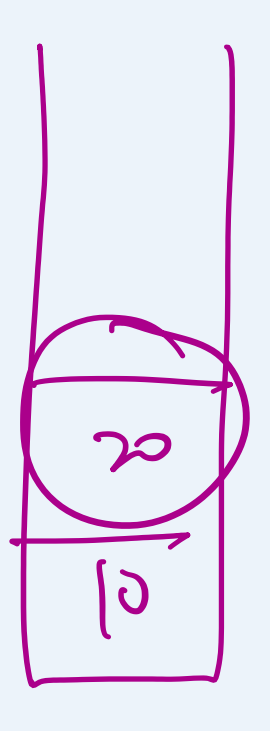


LIFO data structure → Last in first out

operations

1. push (adding something to stack)
2. pop (removing from stack)

peek(); → 20



Stack<Integer> s = new stack();

Implementations :

1. Arrays ✓
2. Linkedlist ✓
3. Queues

1. Arrays

→ | 20 | 40 | 50 | 60 | 70 | ↓ top

top [points to the topmost ele in stack]

Stack overflow

Stack underflow

s.push(20);
s.push(30);
s.pop();
s.push(40);
s.push(50);
s.push(60);
s.push(70);
s.push(80);

Pros

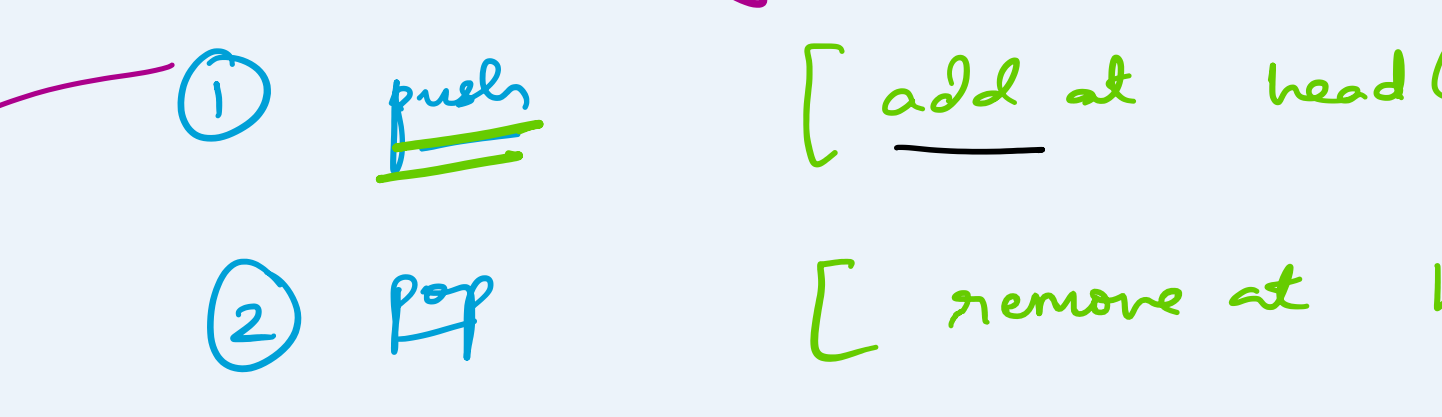
→ Easy to implement.

→ push / pop → O(1), O(1)

→

Cons

→ × dynamic

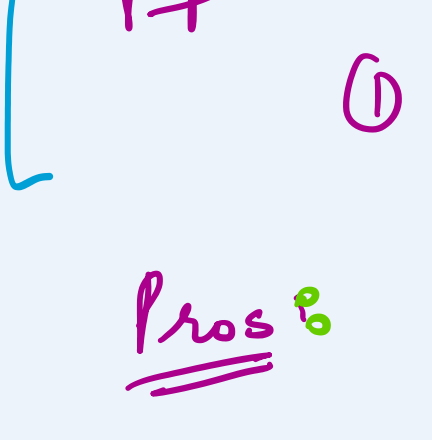


pop();

3

Stack using Linkedlist

1. push [add at head()]
2. pop [remove at head]



push(10);
push(20);
push(30);
push(40);
push(50);
pop();
pop();
push(60);
pop();
pop();

push — O(1)
1. new node
2. new node → head
3. head → new node

pop — O(1)
1. head = head.next.

Pros

Cons

```
push() {
    addAtHead();
}

pop() {
    removeAtHead();
}
```

Ques: Given a string (s) of just containing the characters '(', ')', '{', '}', '[', ']' and determine if the string is valid.

1. open brackets must be closed by same type of brackets
2. open bracket must be closed in correct order

s = ()[]{} true
s = [(] false
s = ()[{()}] true
s = ((()] false

([{

( )

(( ) (( ) ( ) ( )))

left = 1 2 1 2 3 2 3 2 3 2 1 0

if (left == 0)

( ))

left = 1 0

[({]

expression

{ [[ ] { [ ] } ( ) ( )] }

{ [

Bracket opened **last will be closed first**

stack → LIFO

Q Given a stack, reverse it.

I → | 10 | 20 | 30 | 40 | 50 |

O → | 50 | 40 | 30 | 20 | 10 |

× while, for loop

✓ peek(), push, pop

Recursion

1. insertAtBottom()

insertAtBottom(30)

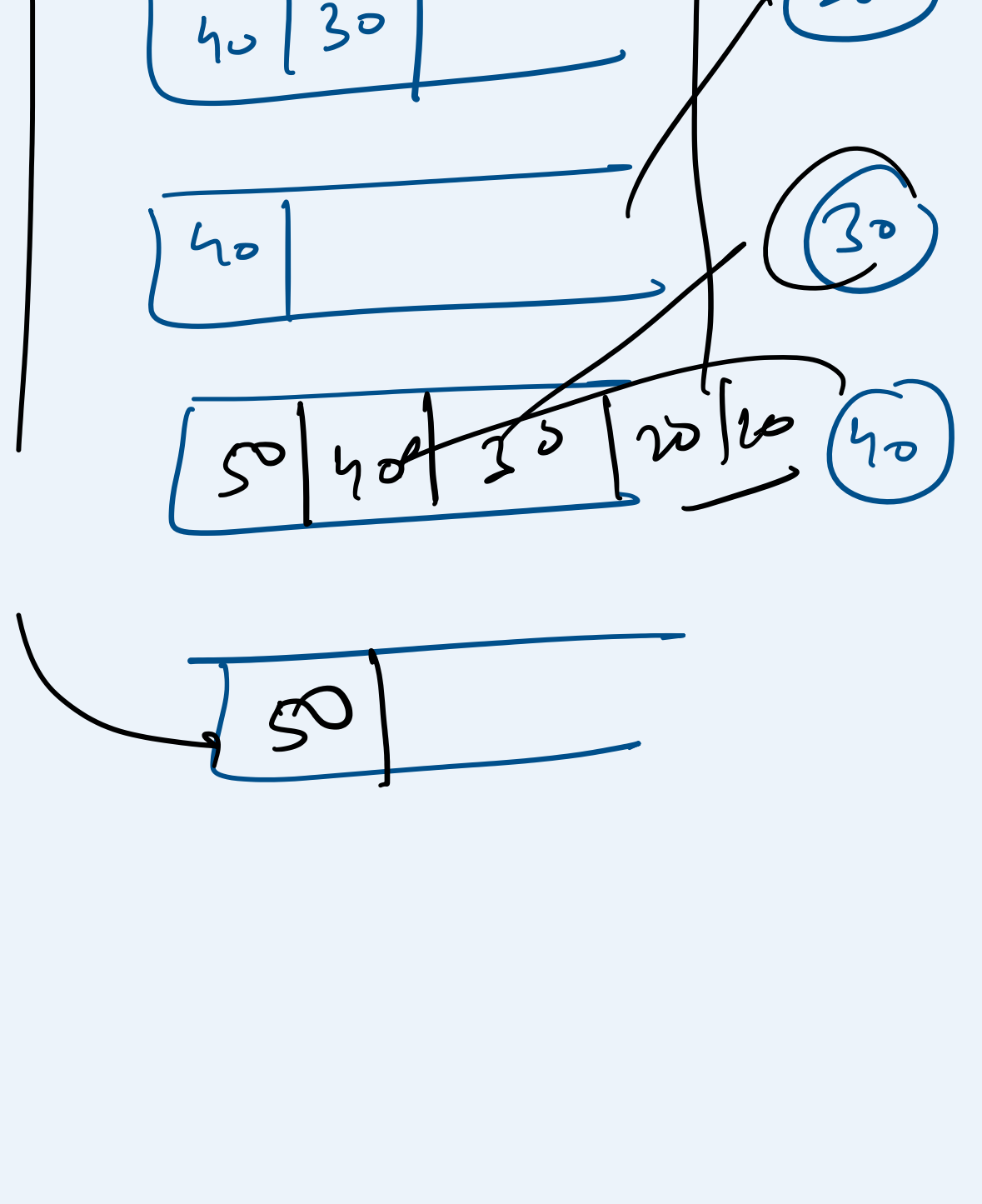
| 10 | 50 | 90 | 40 |

| 30 | 10 | 50 | 90 | 40 |

| 50 | 40 | 30 | 20 | 10 |

reverse()

ch = 50 s.pop()
reverse()
IAB(50);




```
reverse() {
    if (s == empty) {
        return;
    }
    ch = s.pop();
    reverse();
    InsertAtBottom() ;
}

InsertAtBottom(x) {
    if (s == empty) {
        s.push(x);
    }
    else {
        ch = s.pop();
        IAB(x);
        s.push(ch);
    }
}
```