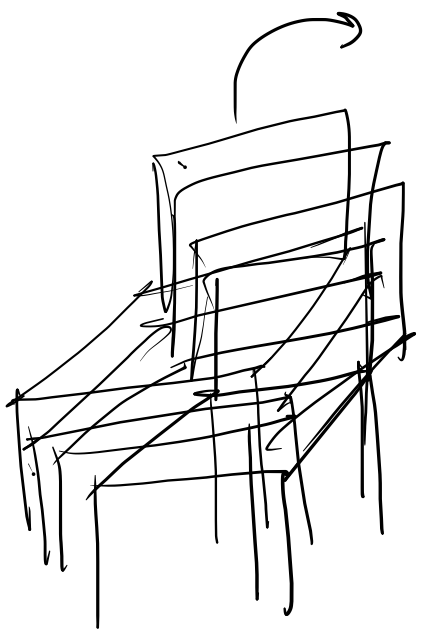
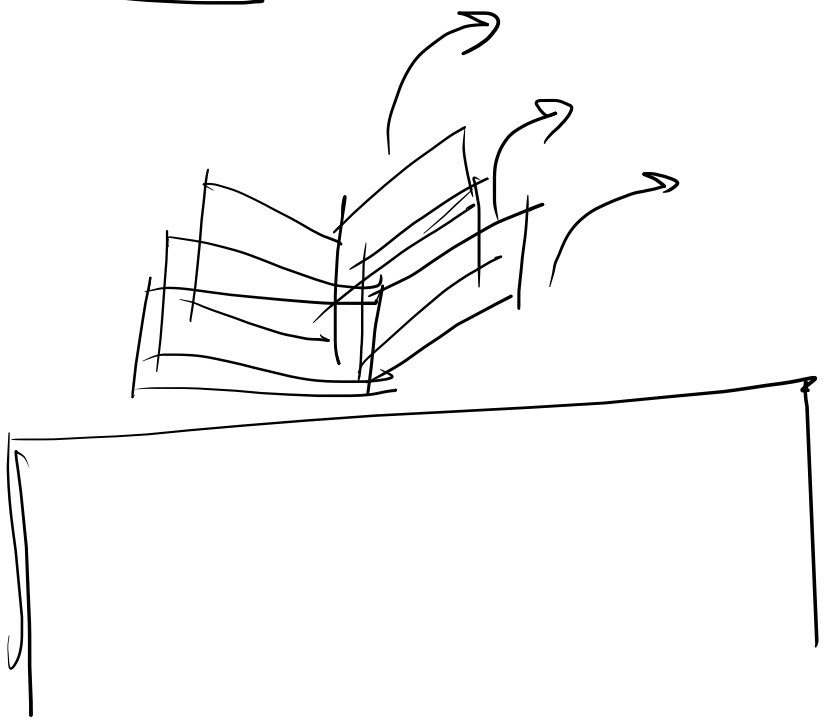
Stack

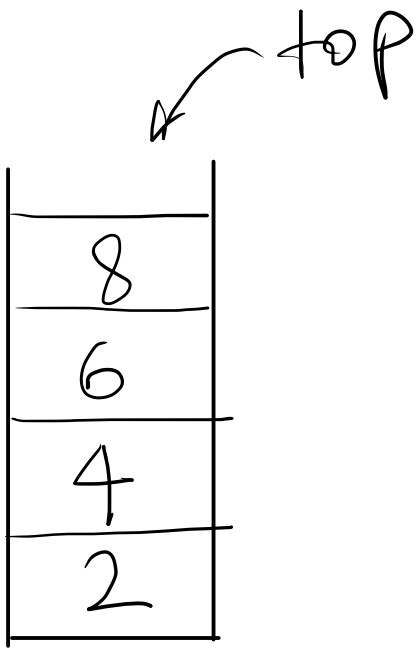


Last In, first out

Stack



LIFO

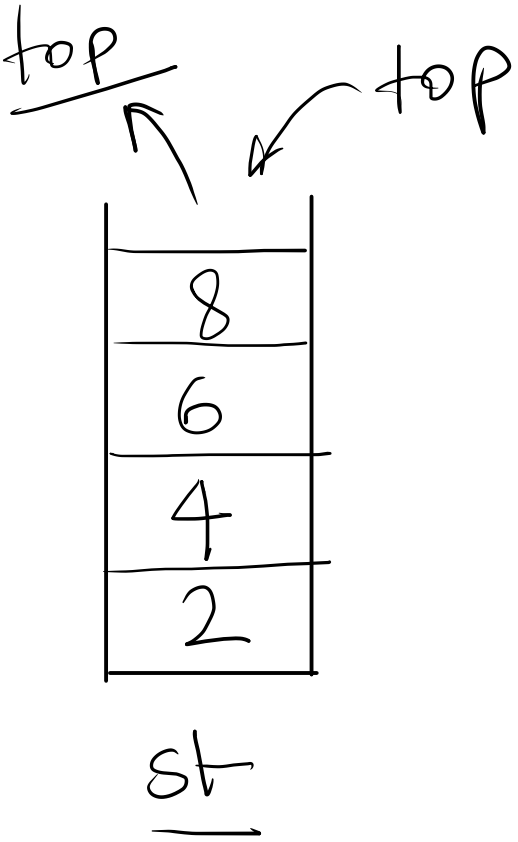


st

Insertion

Push

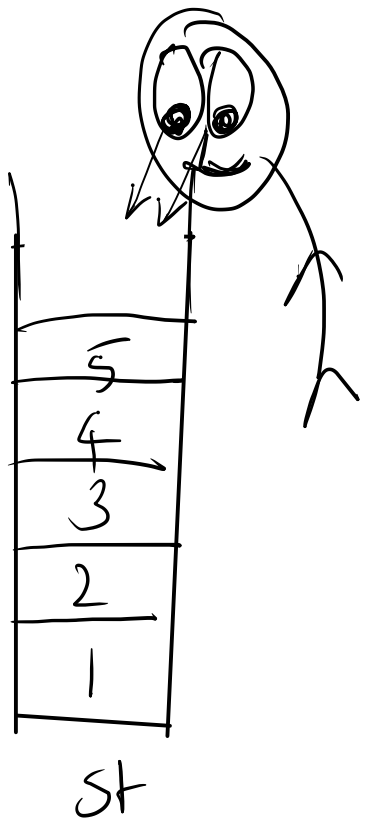
st.push(8)



Deletion

↳ pop()

st.pop()



topmost element

↳ peek

st.peek() → 5

Stack is Full (can't push)

↳ Stack Overflow

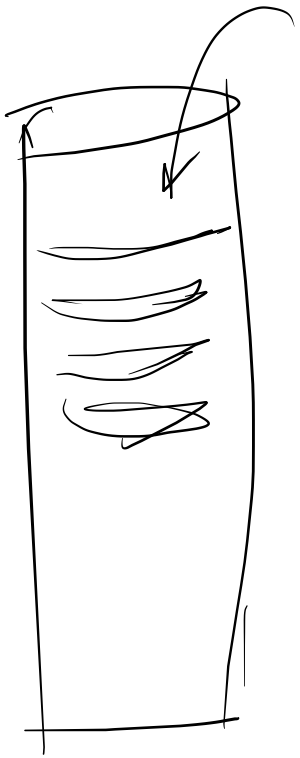
Stack is Empty (can't pop)

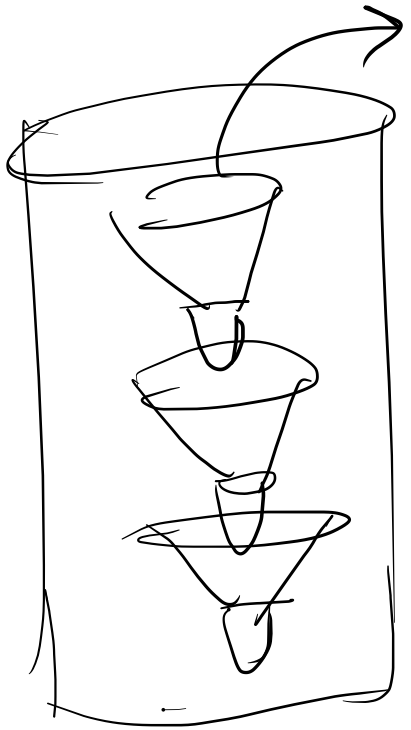
↳ Stack underflow

Stack

↳ size()

st.size() → 3





Stack

Input : [{ ( ) }]

(valid)

Output : true

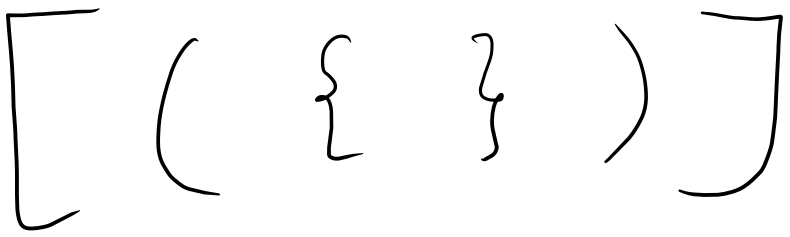
Input : [( ])

Output : false

[( )]

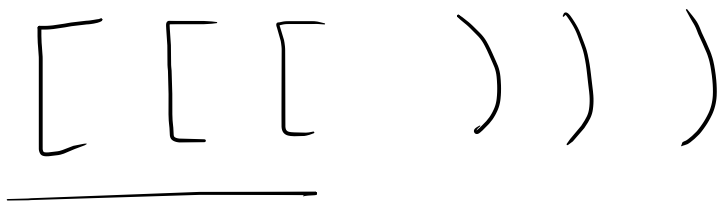
---

true



open = 3

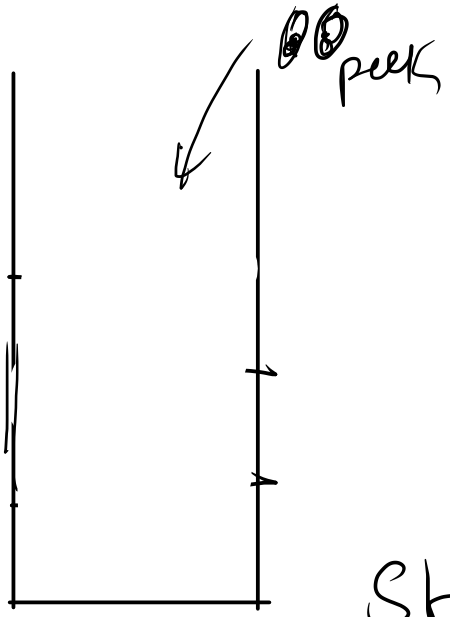
closing = 3



open = 3 close = 3

not valid

S = " [{ ( ) }] "



Opening → push

Closing → check and pop

Stack empty

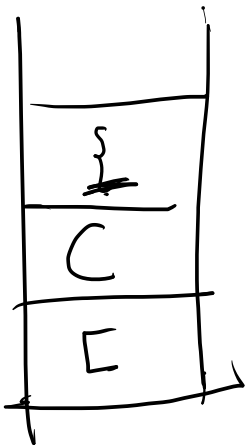
i i i i

[({]) }

Checking failed
return false

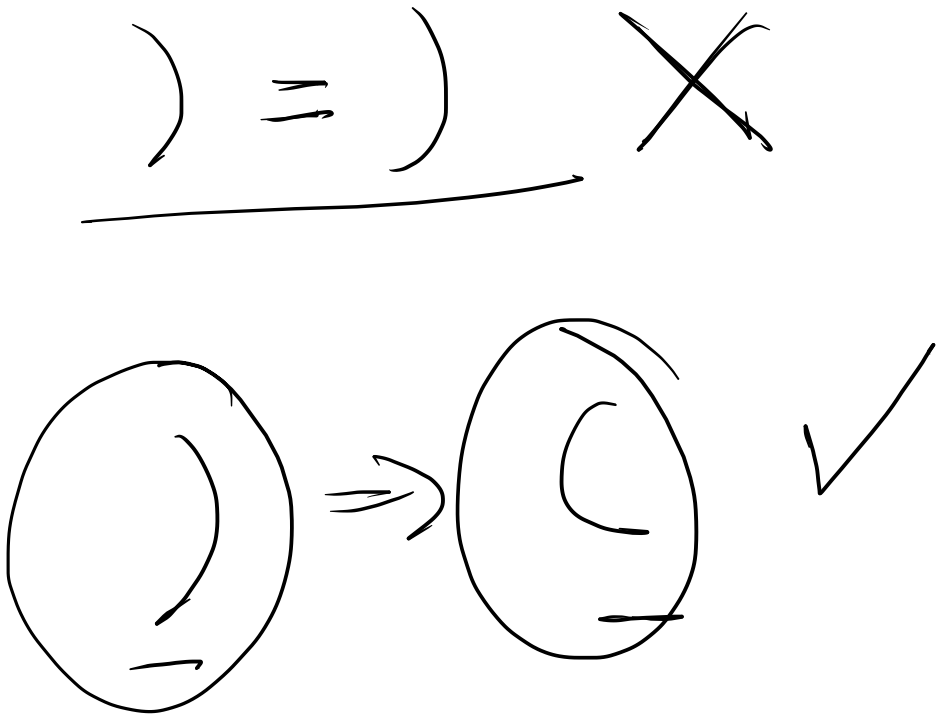
opening → push

closing → check
& pop



T.C = O(n)

Aux space = O(n)

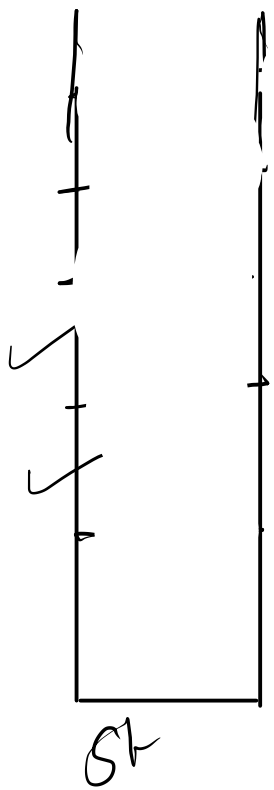


AA cd HD

AA cd D

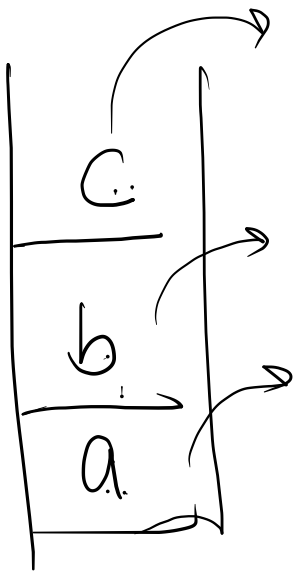
Output AAC

AA cdhH D



am = pop + ams / A + AC

am = AAC



ans = ""

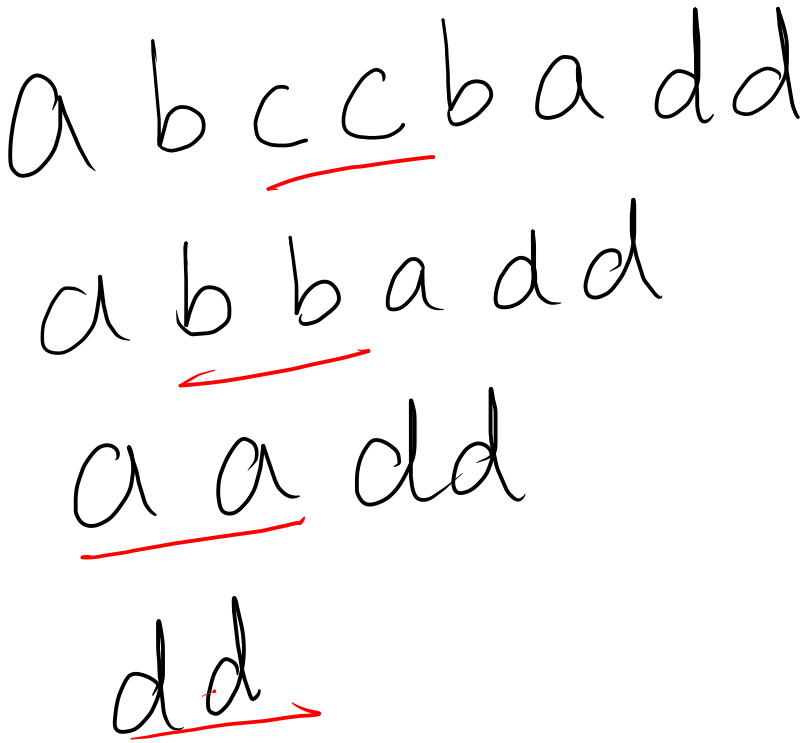
ans = St.pop() + ans

b + c = bc

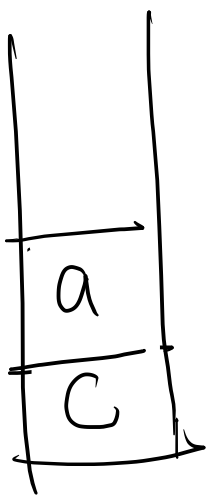
a + bc = abc

T.C = O(n)

Aux S.C = O(n)


a b c c b a d d
a b b a d d
a a d d
d d

a b b a c a

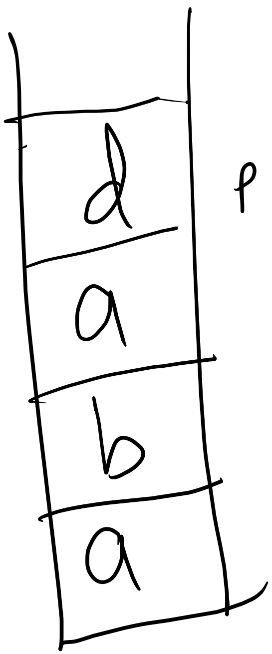


ans = pop + ans

ans = 'c' + 'a'

ans = ca

abaccd



abad

$$TC = O(n)$$

$$\text{Aux } SC = O(n)$$