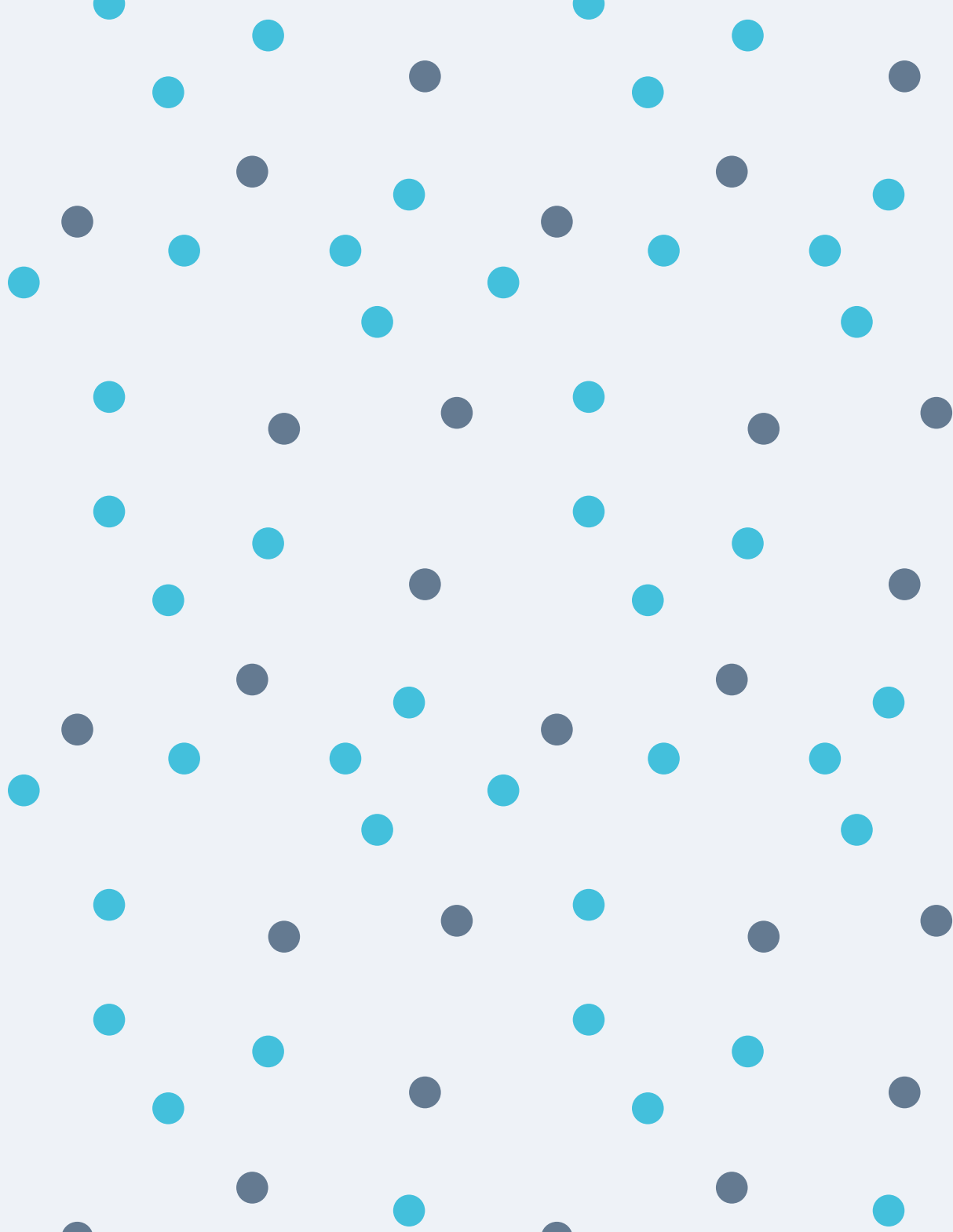
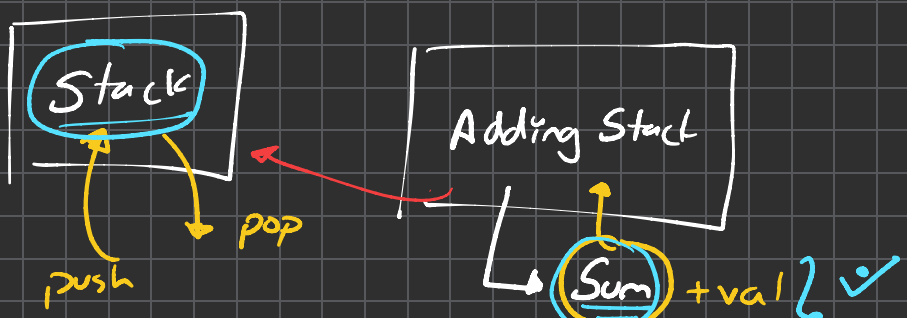
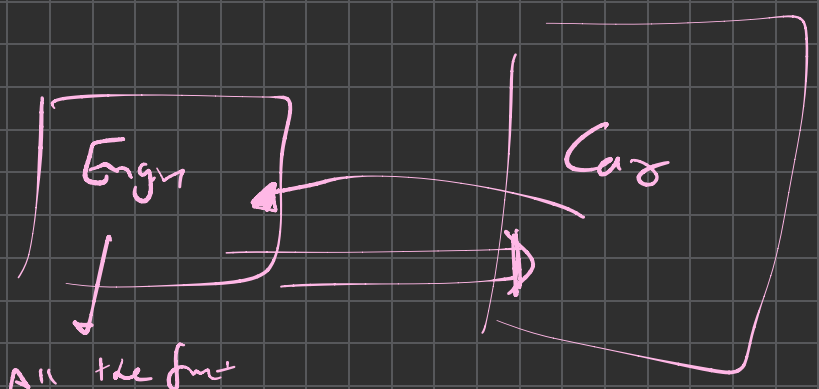




Self.__sum += val

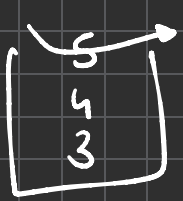
Stack.push(self, val)

push and pop



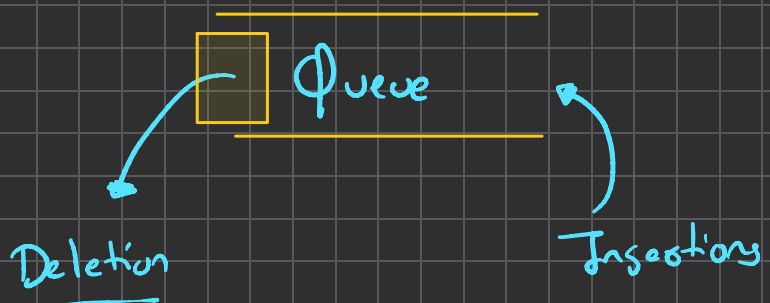
Engine.start()

LIFO

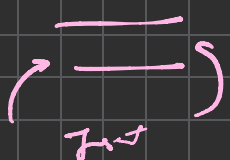


LIFO

Data Structure - Queue ⇒ FIFO



deque →

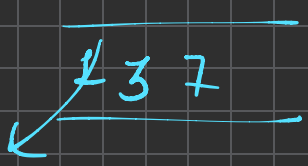


① init

② put → insert

③ get → accessing

↳ popQ



q.put(1) q.put(7)

q.put(3) q.get() ⇒ 1

Balanced Parenthesis

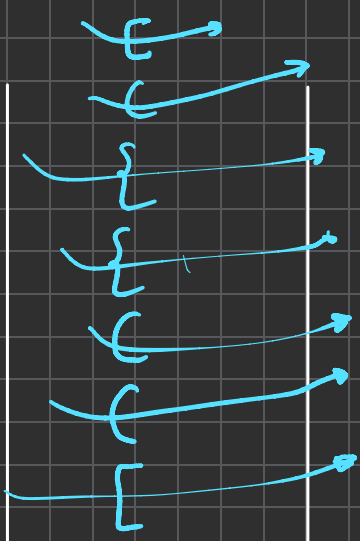
( ), [ ], { }

s = "[()()[{()}][]]"

s = "[][(())]({})"

Find an approach to figure out if the given string is a balanced string or not.

s = "[()()[{()}][]]"

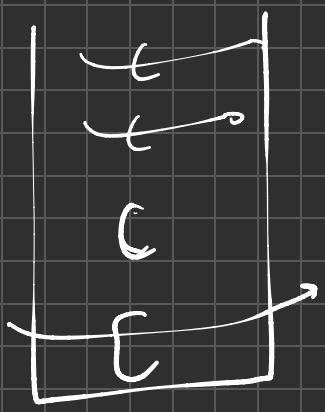


if ch = opening bracket
↳ Take and put it in stack.

if ch = closing bracket
↳ Check correct pair
↳ Remove from stack

if at the end, stack is empty.

s = " [ ] [(( ))] (} "



Not Balanced

Approach:

1. Iterate through each character in the string.
2. If ch == '(' or ch ==']' or ch =='{'
   s.push(ch)
3. If (ch == ')' and s.top == '(') or (ch == ']' and s.top == '[') or (ch == '}' and s.top == '{') :
   s.pop()
4. NOT BALANCED. Exit from the function

5. At last, after completing the iteration, if the stack remains empty, It is a balanced parenthesis.