面向对象

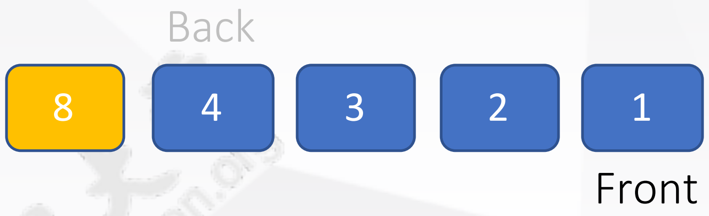
- 面向对象编程

大纲

- 什么是面向对象
- 为什么面向对象

➢ Queue (FIFO) First-In, First-Out First-Come, First-Serve

Put (Enqueue) & Get (Dequeue)



Back

8 4 3 2 1

Front

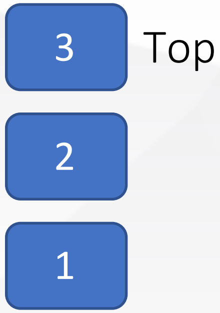
queue.put(8)

item = queue.get()

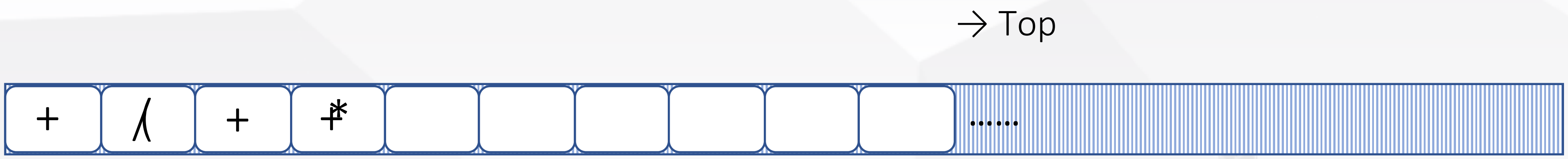
➢ Stack (FILO) First-In, Last-Out

Push & Pop

stack.push(8)
item = stack.pop()



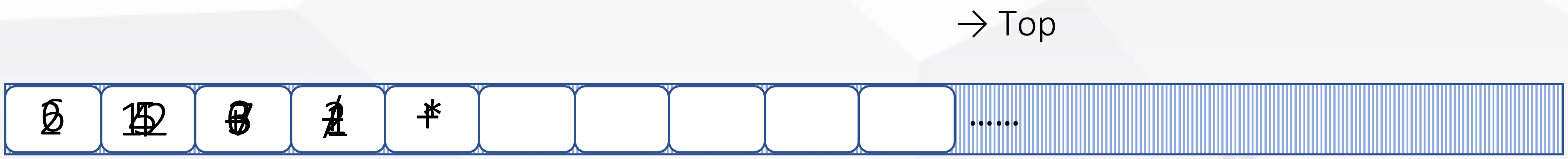
2 + (5 + 3 * 2 + 1) / 3



OUTPUT: 2 5 3 2 * 1 + + 3 / +

1. If the current input token is an **operand**, append it to the output string
2. If the current input token is an **operator**, pop off all operators that have *equal or higher precedence* and append them to the output string; push the operator onto the stack
3. If the current input token is (, push it onto the stack
4. If the current input token is), pop off all operators and append them to the output string until a (is popped; discard the (.
5. If the end of the input string is found, pop all operators and append them to the output string.

OUTPUT: 2 5 3 2 * 1 + + 3 / +

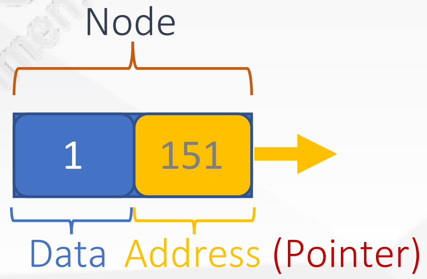


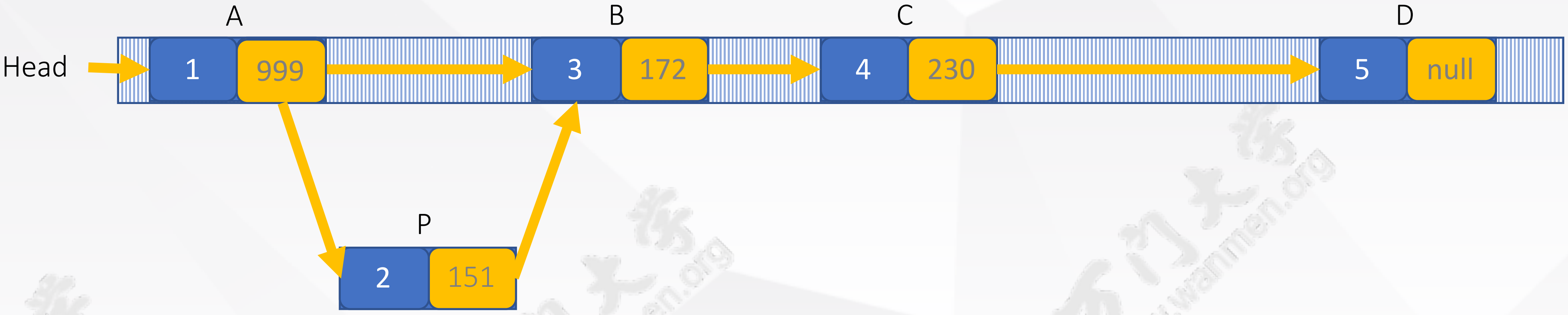
RESULT = 6

1. Scan the expression left to right
2. If **operand**, push it into stack
3. When an **operator** is found, apply the operation to the preceding two operands
4. Replace the two operands and operator with the calculated value (three symbols are replaced with one operand)
5. Continue scanning until only a value remains––the result of the expression

Linked List

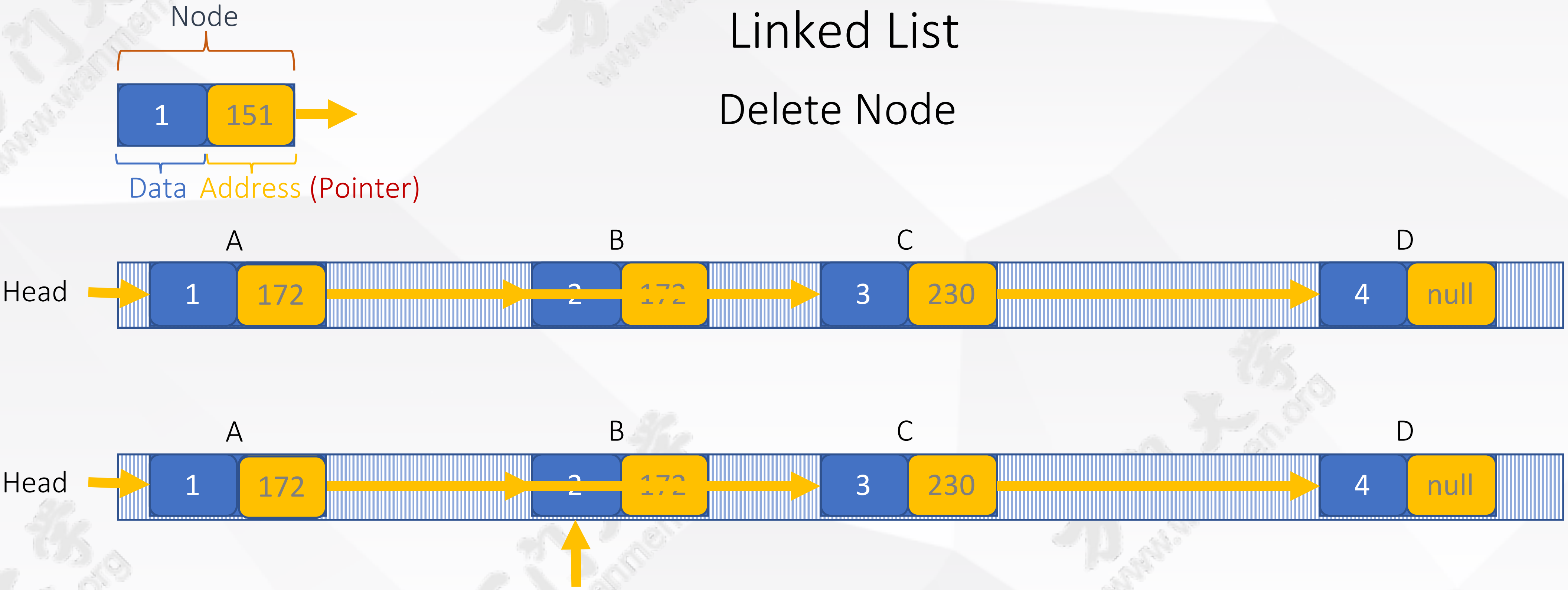
Add Node



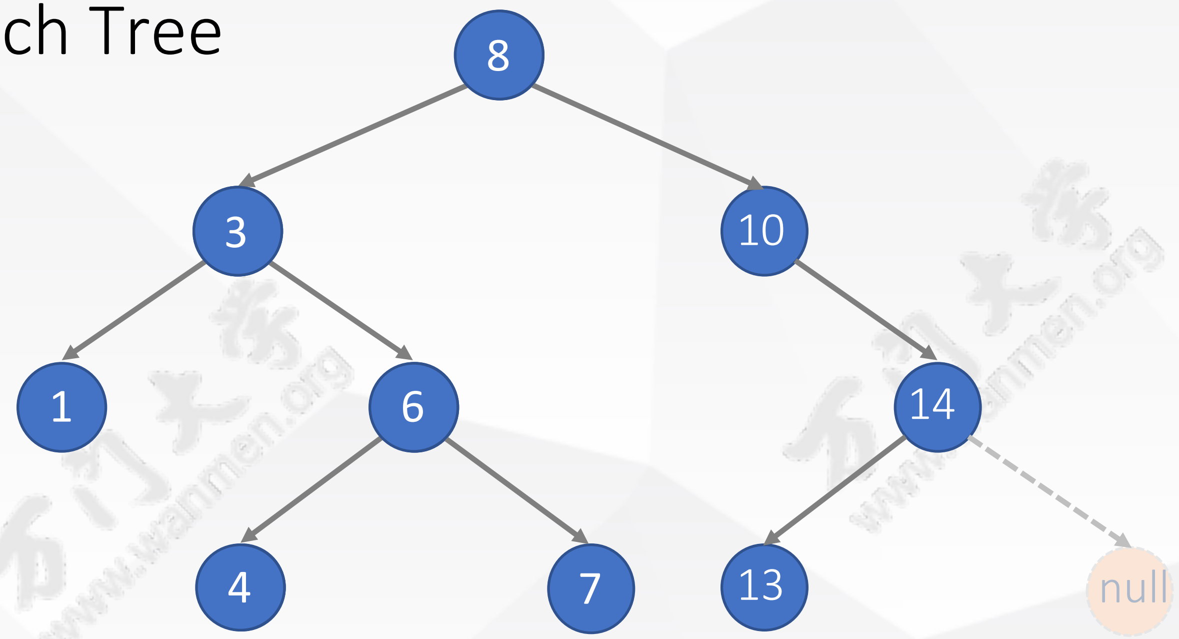


Linked List

Delete Node



➢ Tree: Binary Search Tree



Find 4

Find 15

Average Search / Insert / Delete Time $\Theta\ (\log n)$

Coding