问题解答

Python的类继承实例分析

两周内容小结：W03/04：基本结构

关于H3作业

数据结构与算法（Python）-07/0313

陈斌 gischen@pku.edu.cn 北京大学地球与空间科学学院

目录

- 问题解答
- Python的类继承实例分析
- 两周内容小结

  W03/04：基本结构
- 关于H3作业

问题解答

› 1 输入类中的操作时忘记打括号为什么不报错？

```
def func():
```

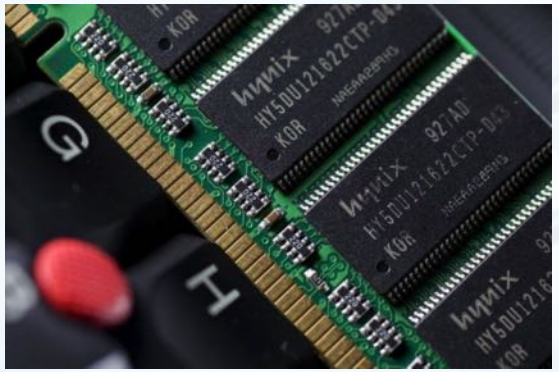
› 2

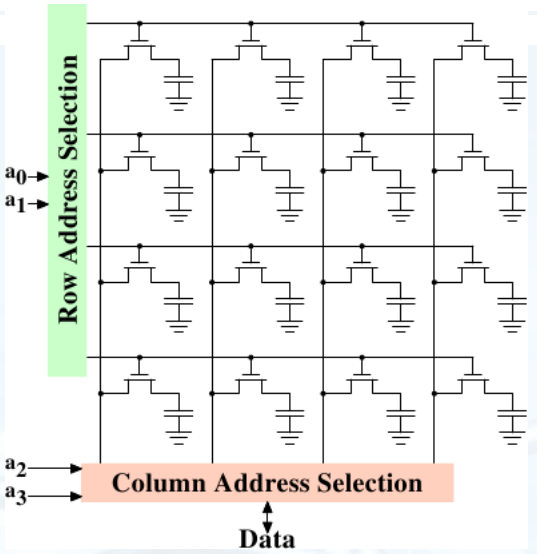
```
def startNext(self,newtask):
    self.currentTask = newtask
    self.timeRemaining = newtask.getPages() \
                         * 60/self.pagerate
```

打印新作业

问题解答

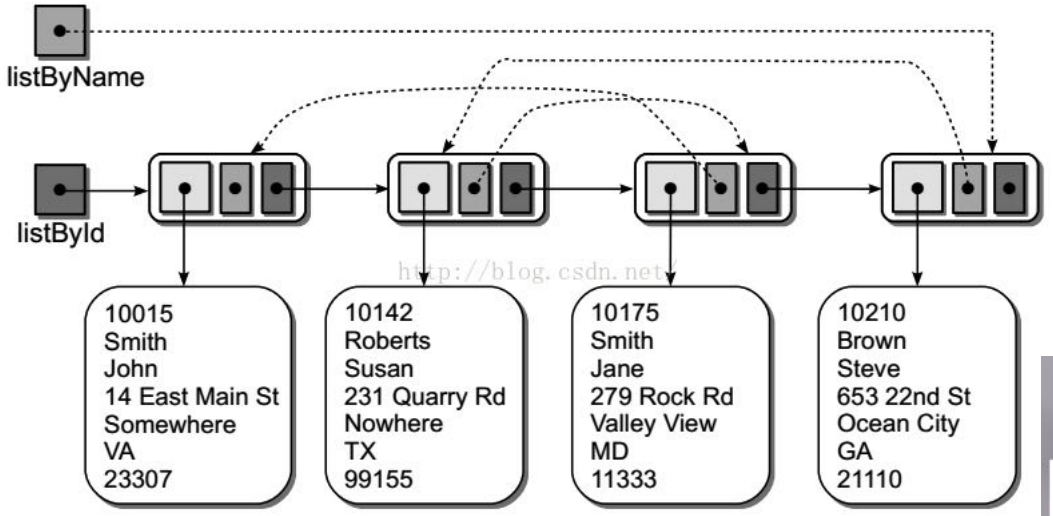
3 请问链表顺序存储、顺序表随机存储是什么意思？





问题解答

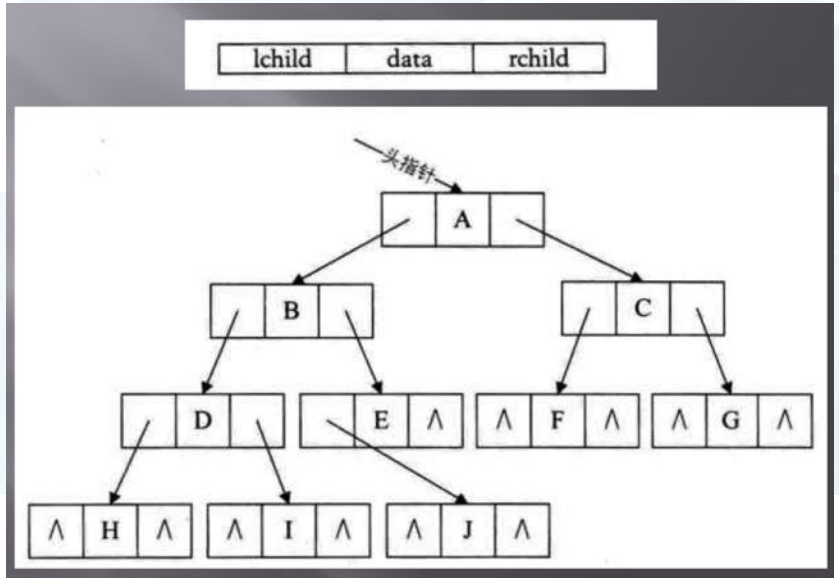
请问什么是多重链表、二叉链表和指针域



多重链表

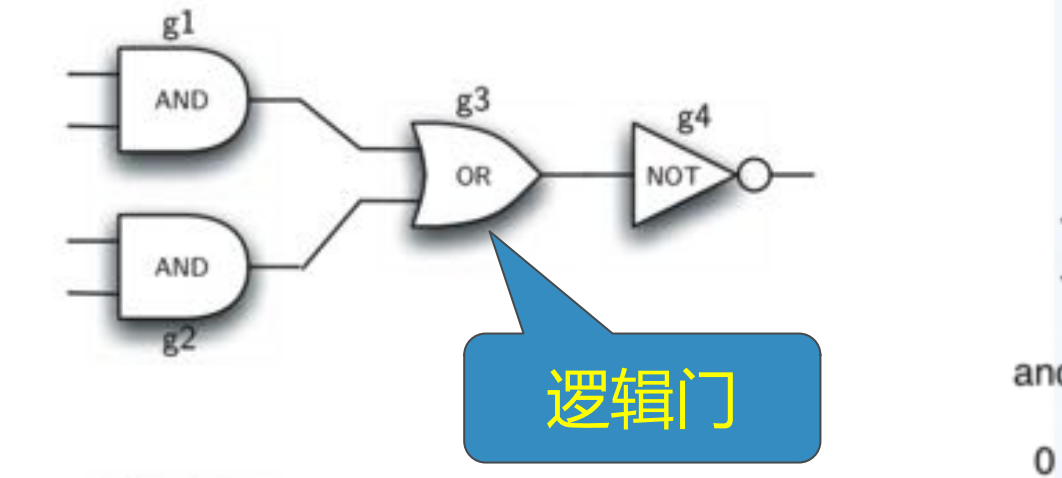
```
1  # 多重链表的结点
2
3  class StudentMListNode(object):
4      def __init__(self, data):
5          self.data = data
6          self.nextById = None
7          self.nextByName = None
```

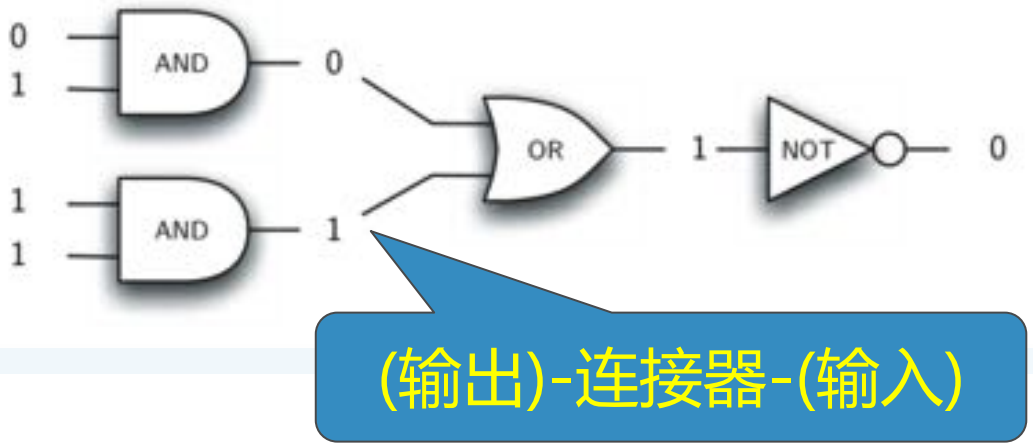
二叉链表



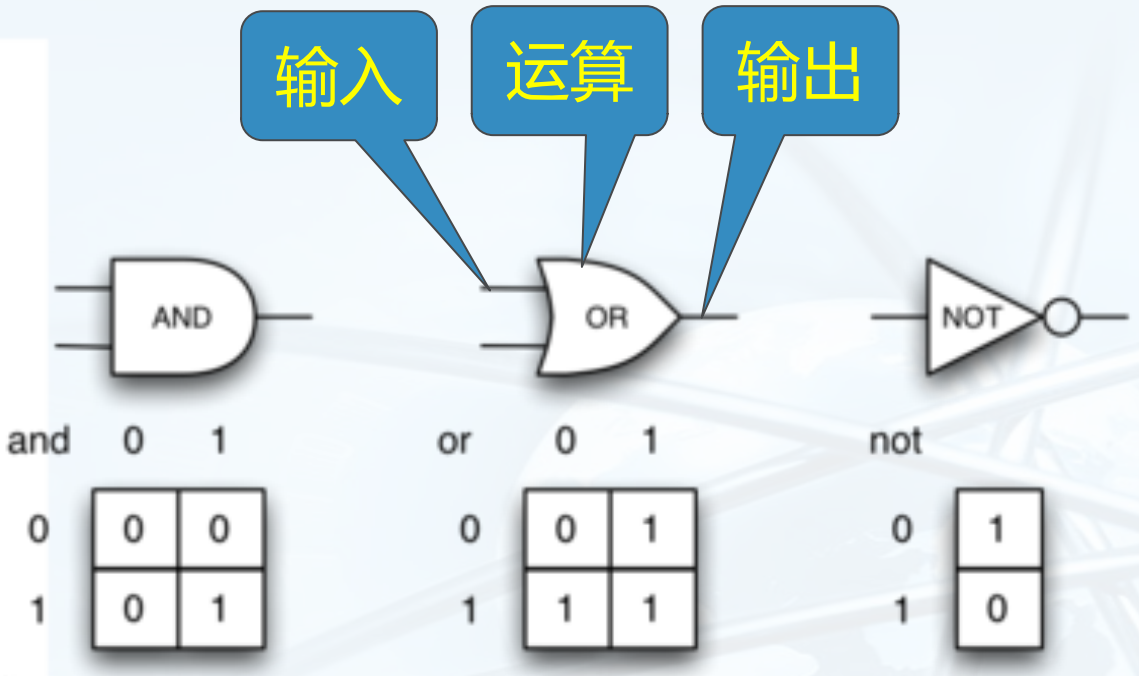
Python的类继承实例分析

模拟一个逻辑门构成的电路

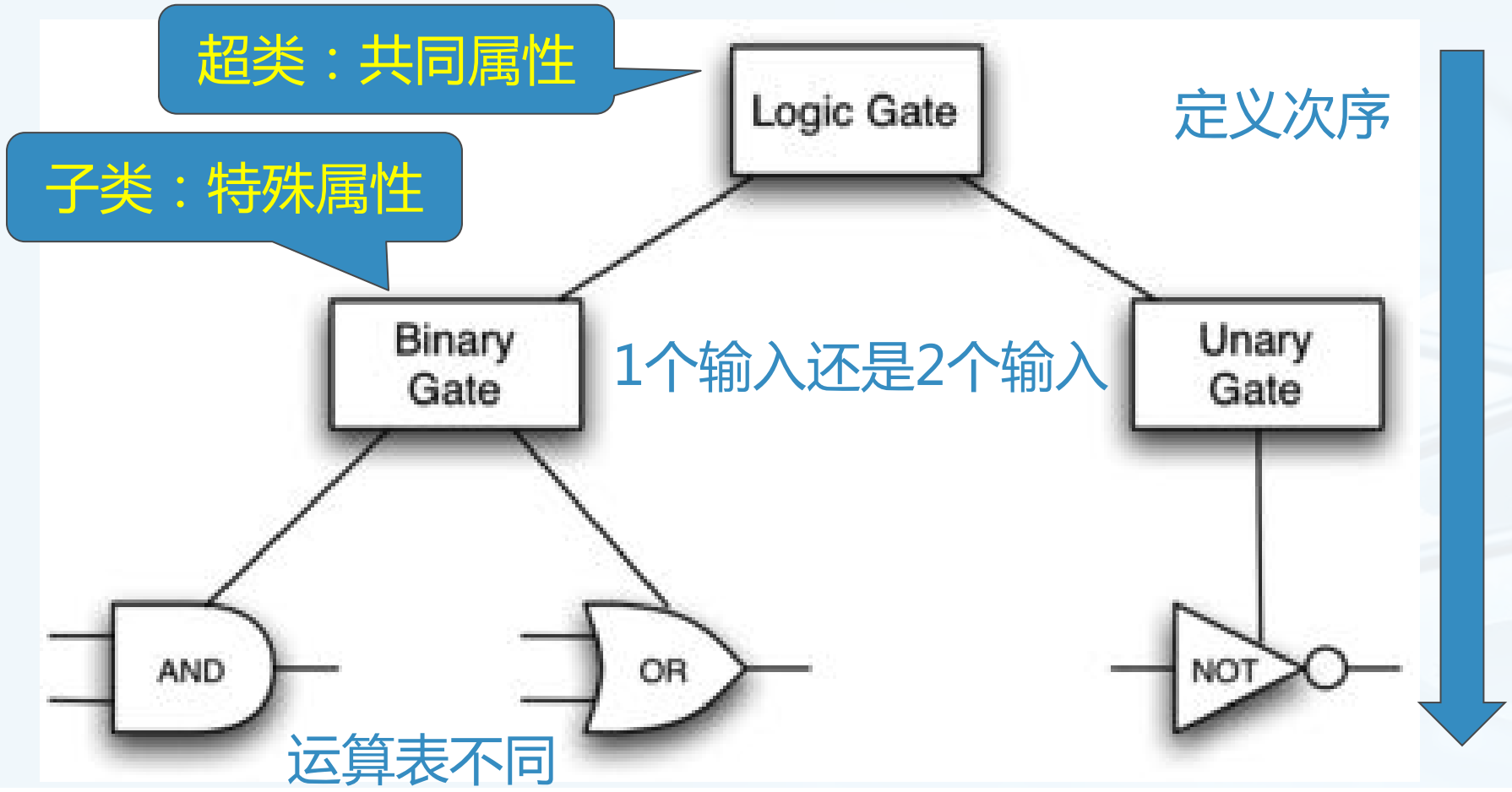




共同属性



类层次图：继承关系


超类：共同属性
子类：特殊属性
Logic Gate
定义次序
Binary Gate
1个输入还是2个输入
Unary Gate
AND
OR
NOT
运算表不同

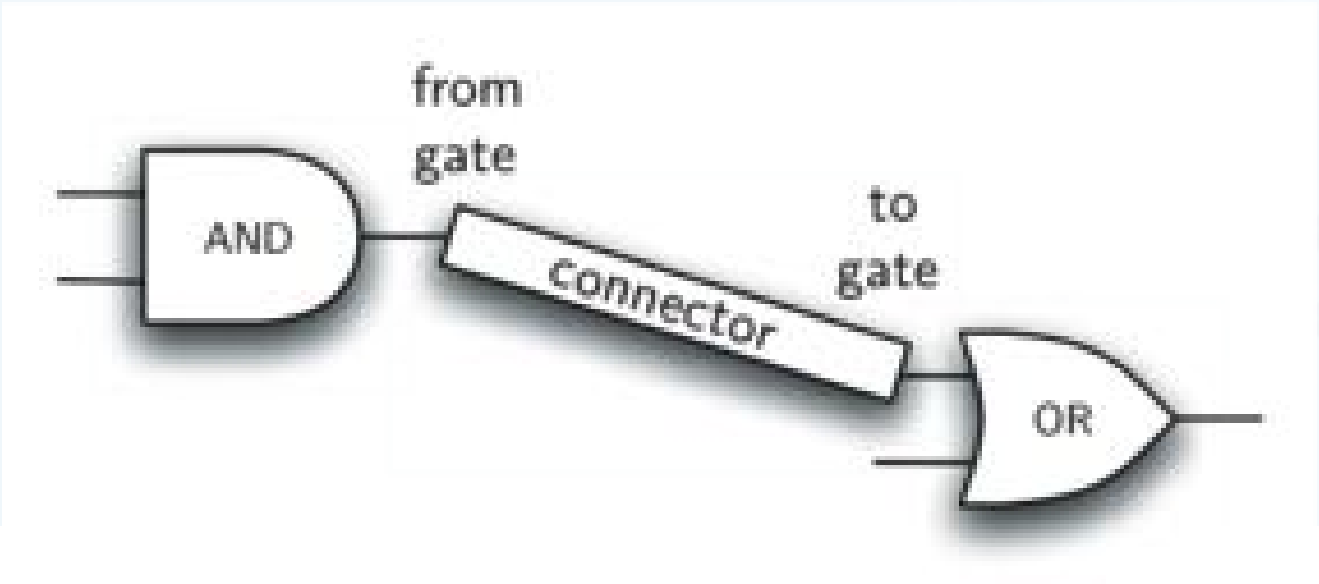
连接器Connector

一个连接器对象

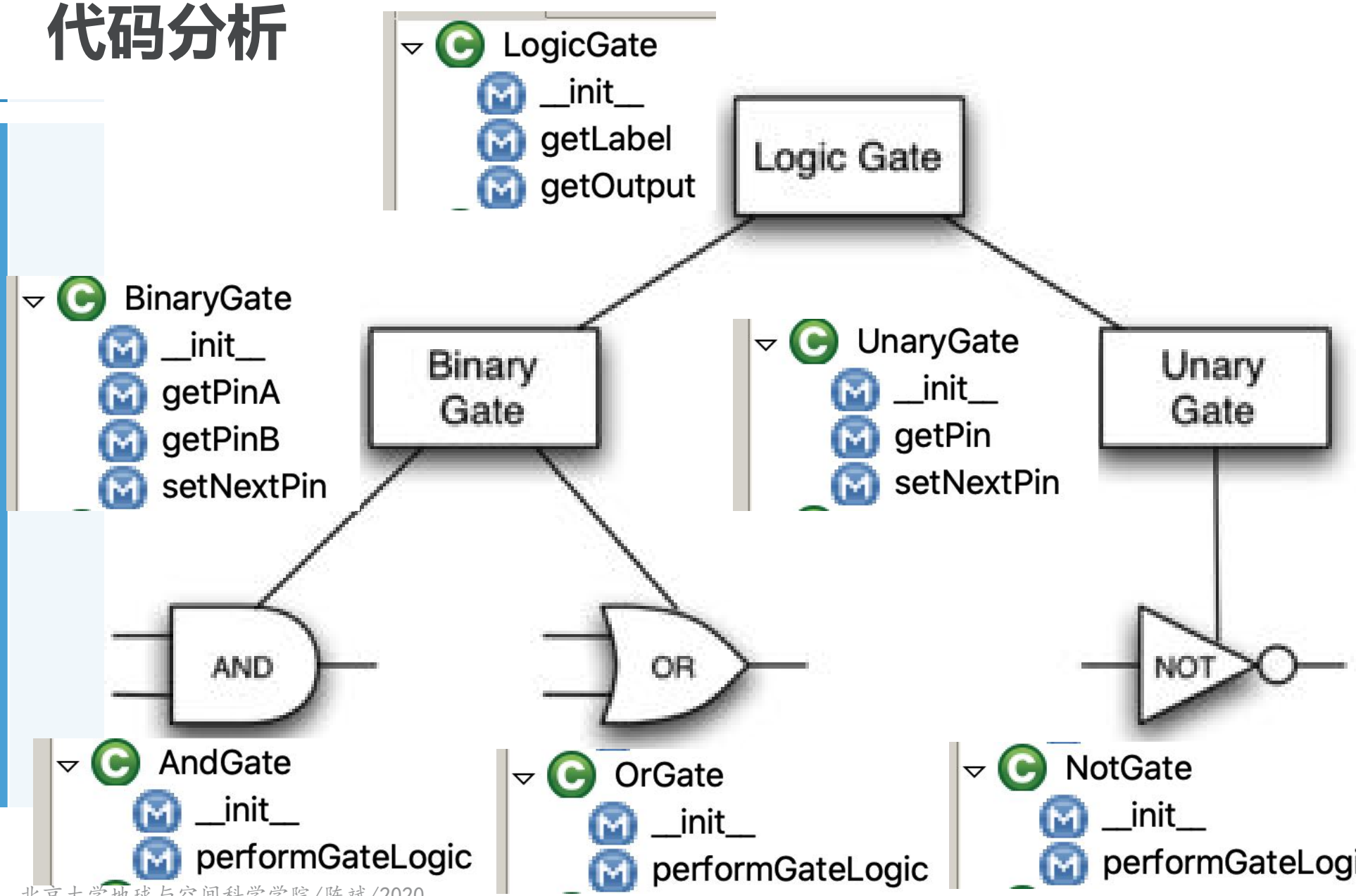
连接了fromgate的输出和togate的输入

所以一个逻辑门的输入有两种情况：

- 最上游的手工输入指定，
- 或者作为下游逻辑门通过连接器的fromgate的输出

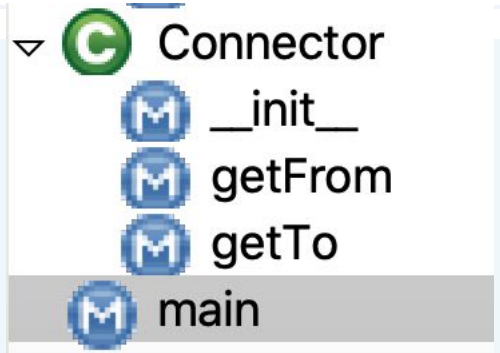


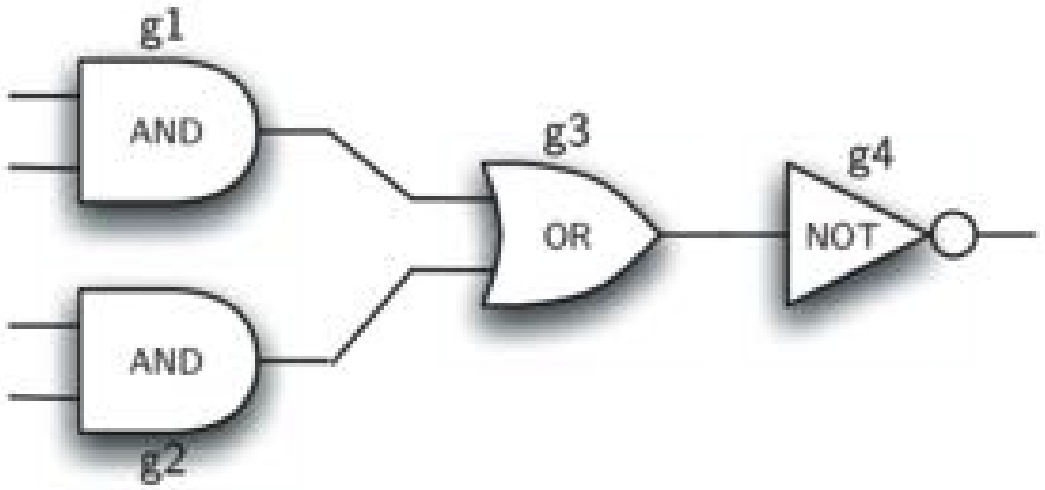
代码分析



Canvas有链接

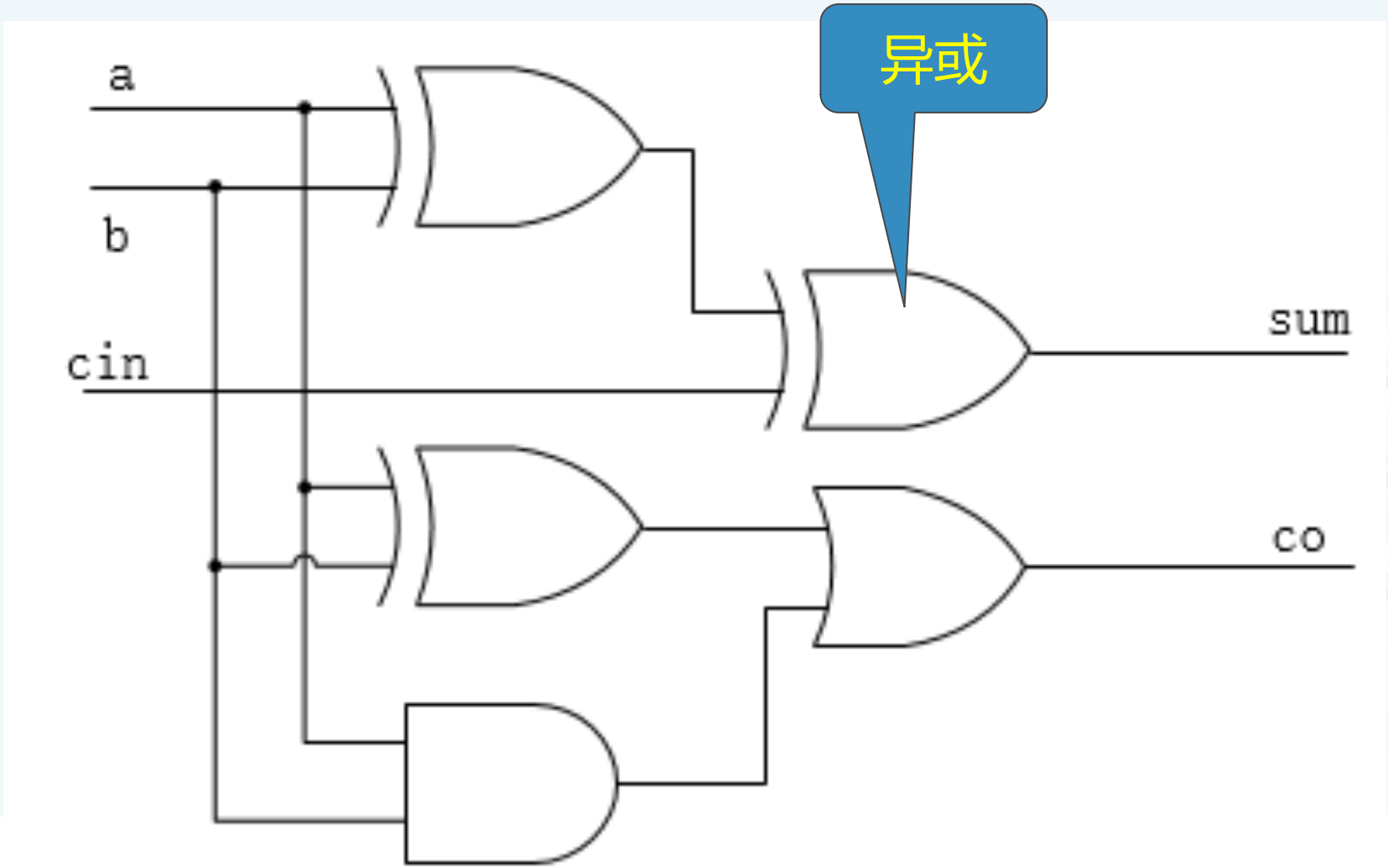
连接器和建立电路






```
119 def main():
120     g1 = AndGate("G1")
121     g2 = AndGate("G2")
122     g3 = OrGate("G3")
123     g4 = NotGate("G4")
124     c1 = Connector(g1,g3)
125     c2 = Connector(g2,g3)
126     c3 = Connector(g3,g4)
127     print(g4.getOutput())
128
129 main()
```

练习：做一个1位全加器？



本章目标

- 了解**抽象数据类型**：栈**stack**、队列**queue**、双端队列**deque**和列表**list**；
- 能够采用Python列表数据类型来**实现**stack/queue/deque等抽象数据类型；
- 了解基本线性数据结构各种具体实现算法的**性能**；
- 了解**前缀**、**中缀**和**后缀**表达式；
- 采用stack对后缀表达式进行**求值**；
- 采用stack将中缀表达式**转换**为后缀表达式；
- 采用queue进行基本的点名报数**模拟**；
- 能够识别问题属性，选用stack、queue或者deque中更为**合适的**数据结构；
- 能够通过节点和节点引用的模式，采用**链表**来实现抽象数据类型list；
- 能够**比较**链表实现与Python的list实现之间的算法**性能**。

W03/04：基本结构

- 301 什么是线性结构
- 302 栈抽象数据类型及Python实现
- 303 栈的应用：简单括号匹配
- 304 栈的应用：十进制转换为二进制
- 305/306 表达式转换
- 307 后缀表达式求值

W03/04：基本结构

- 308 队列抽象数据类型及Python实现
- 309 队列的应用：热土豆
- 310/311 队列的应用：打印任务
- 312 双端队列抽象数据类型及Python实现+回文词判定
- 313 无序表抽象数据类型及Python实现
- 314 无序表的链表实现
- 315 有序表抽象数据类型及Python实现
- 316 线性结构小结

关于本周作业

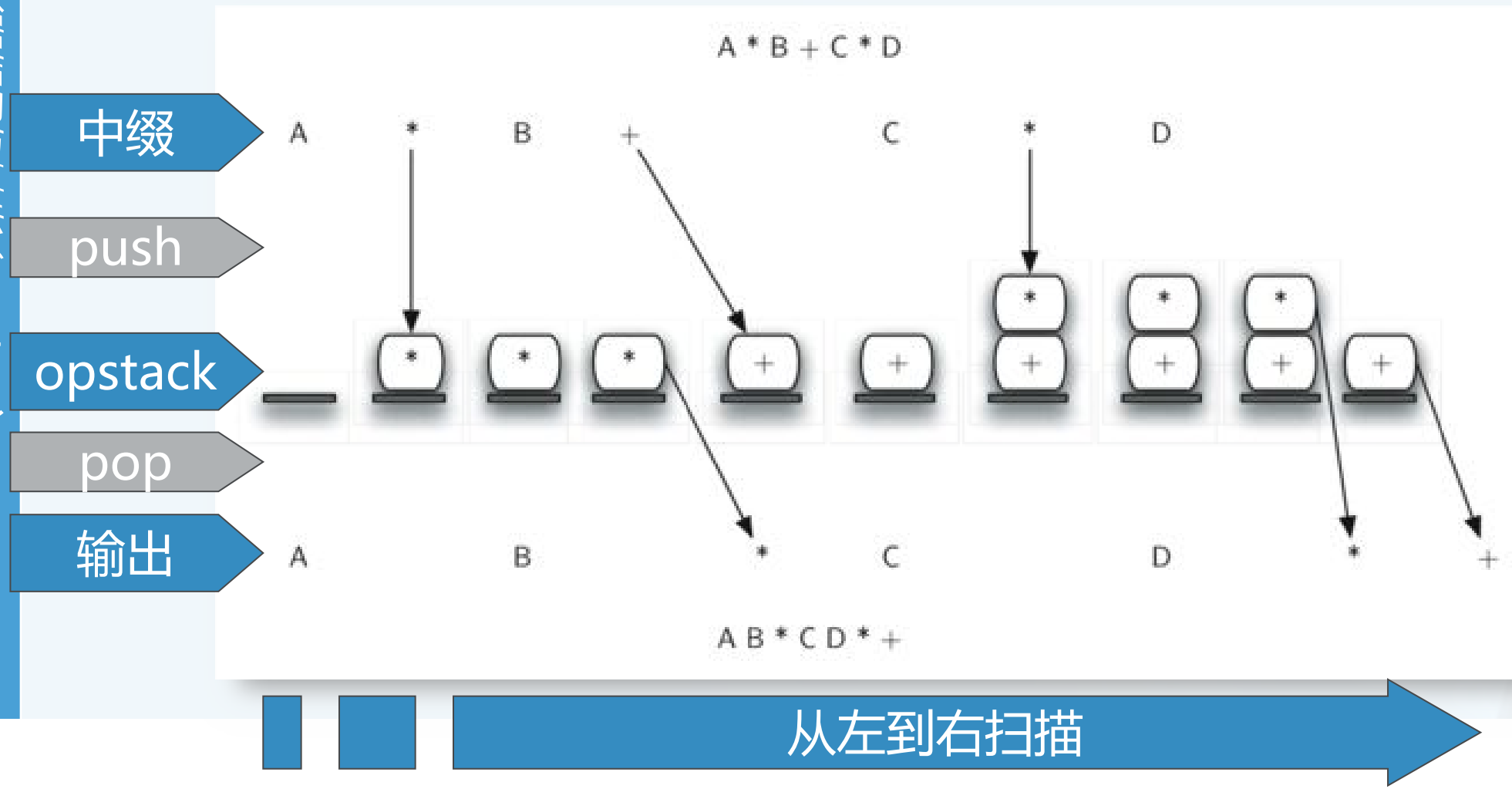
- **必做：见Canvas页面和gis4g公告**
  慕课在线测验，和【H3】栈与队列编程作业
- **选做：慕课的OJ作业**
- **【H3】作业提交PyLn平台（见gis4g网站）**
- **关于PyLn平台可以调用的库**

【H3】栈与队列编程作业

- H3-1 中缀表达式求值；
- H3-2 基数排序；
- H3-3 HTML标记匹配；
- H3-4 链表实现栈和队列；
- H3-5 双链无序表。

- DDL 3月18日 18点

通用的中缀转后缀算法：实例



代码

```
from pythonds.basic.stack import Stack

def infixToPostfix(infixexpr):
    prec = {}
    prec["*"] = 3
    prec["/"] = 3
    prec["+"] = 2
    prec["-"] = 2
    prec["("] = 1
    opStack = Stack()
    postfixList = []
    tokenList = infixexpr.split()

    for token in tokenList:
        if token in "ABCDEFGHIJKLMNOPQRSTUVWXYZ" or token in "0123456789":
            postfixList.append(token)
        elif token == '(':
            opStack.push(token)
        elif token == ')':
            topToken = opStack.pop()
            while topToken != '(':
                postfixList.append(topToken)
                topToken = opStack.pop()
        else:
            while (not opStack.isEmpty()) and \
               (prec[opStack.peek()] >= prec[token]):
                  postfixList.append(opStack.pop())
            opStack.push(token)

    while not opStack.isEmpty():
        postfixList.append(opStack.pop())
    return " ".join(postfixList)
```

记录操作符优先级

解析表达式到单词列表

操作数

(

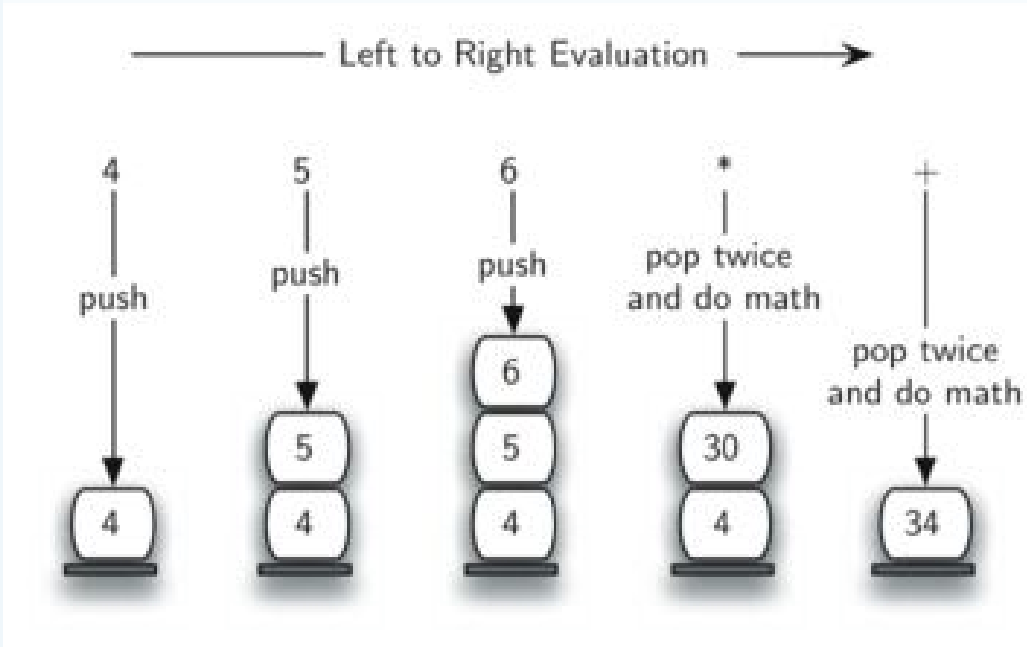
)

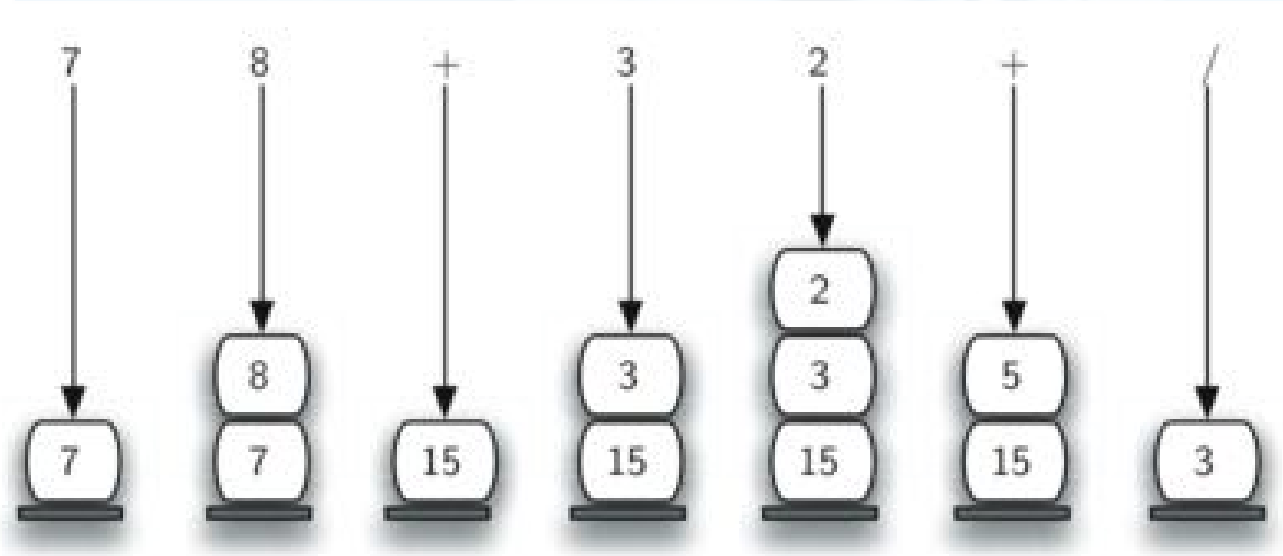
操作符

操作符

合成后缀表达式字符串

后缀表达式求值：实例


Left to Right Evaluation
4
push
5
push
6
push
*
pop twice
and do math
+
pop twice
and do math
4
5
4
6
5
4
30
4
34


7
8
+
3
2
+
/
7
8
7
15
3
15
2
3
15
5
15
3

【H3】栈与队列编程作业

- **建立10个可用下标访问的队列**

```
q = [ Queue() for i in range(10) ]
```

```
q[0]~q[9]
```

- **HTML标记，开标记和闭标记，配对和嵌套**

```
<pre> </pre>
```

```
<html> </html>
```

【H3】栈与队列编程作业

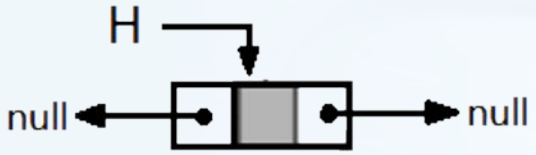
- **建立10个可用下标访问的队列**

  `q = [ Queue() for i in range(10) ]  q[0]~q[9]`

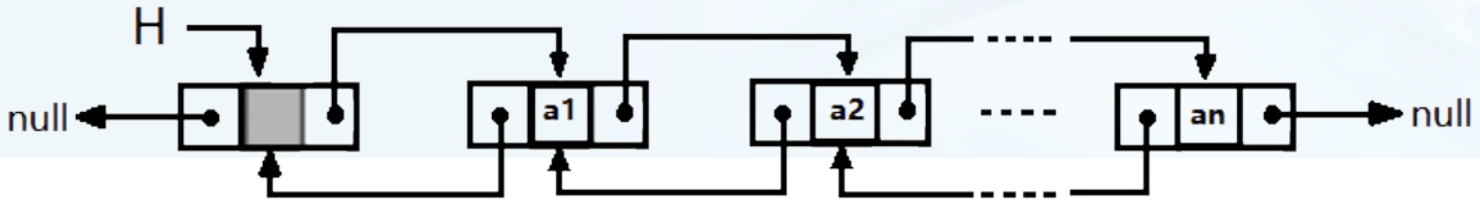
- **HTML标记，开标记和闭标记**

  `<pre> </pre>`

- **单链表和双链表**



(a) 有头结点的 双链空表



(b) 有头结点的 双链表