三、基本结构（上）

P15：线性结构

1. 线性结构是一种有序数据项的集合，其中每个数据项都有唯一的前驱和后继。
2. 两端的称呼并不是关键， 不同线性结构的关键区别在于数据项增减的方式：有的结构只允许数据项从一端添加，而有的结构则允许数据项从两端移除。
3. 4个最简单但功能强大的结构入手开始研究数据结构：

栈Stack， 队列Queue， 双端队列Deque和列表List；

这些数据集的共同点在于，数据项之间只存在先后的次序关系，都是线性结构。

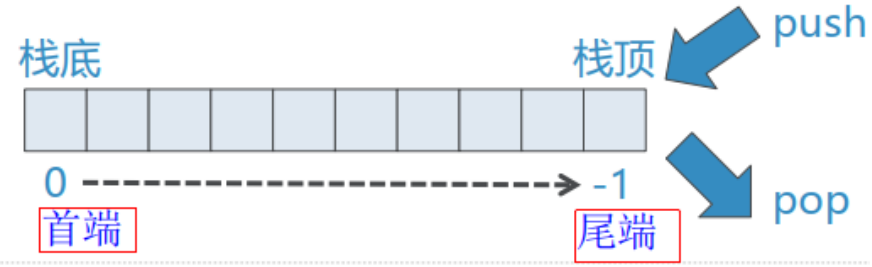
这些线性结构是应用最广泛的数据结构，它们出现在各种算法中，用来解决大量重要问题

P16：？栈的抽象数据类型及其python实现

例子：

1. 文档中的“undo”选项，返回最后操作的一次；
2. 网页中的“Back”选项，返回最后浏览的网页；

1. 定义：



1. 基本操作

❖抽象数据类型“栈”定义为如下的操作

Stack()：创建一个空栈，不包含任何数据项

push(item)：将item加入栈顶，无返回值

pop()：将栈顶数据项移除，并返回，栈被修改

peek()：“窥视”栈顶数据项，返回栈顶的数据项但不移除，栈不被修改

isEmpty()：返回栈是否为空栈

size()：返回栈中有多少个数据项

2. 操作样例

抽象数据类型Stack：操作样例

Stack Operation	Stack Contents	Return Value
s= Stack()	[]	Stack object
s.isEmpty()	[]	True
s.push(4)	[4]	
s.push('dog')	[4,'dog']	
s.peek()	[4,'dog']	'dog'
s.push(True)	[4,'dog',True]	
s.size()	[4,'dog',True]	3
s.isEmpty()	[4,'dog',True]	False
s.push(8.4)	[4,'dog',True,8.4]	
s.pop()	[4,'dog',True]	8.4
s.pop()	[4,'dog']	True
s.size()	[4,'dog']	2

问题：

？？？如何导入模块Class?

？？？如何在另一平台上导入栈?

P17：？简单括号匹配

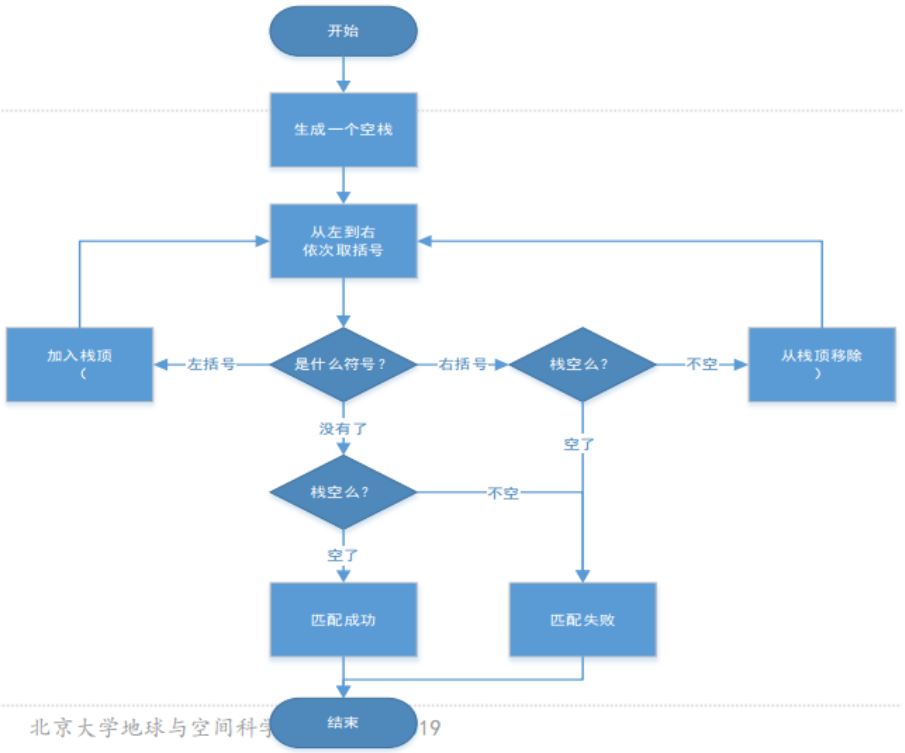
？？？力扣题目的实现?

1. 规则：
1. 首先，每个开括号要恰好对应一个闭括号；
2. 其次，每对开闭括号要正确的嵌套；

例子：正确：(()()()())， ((((()))))，(()((())()))；

错误：((((((())， ()))， (()()(()；

2. 算法流程图：

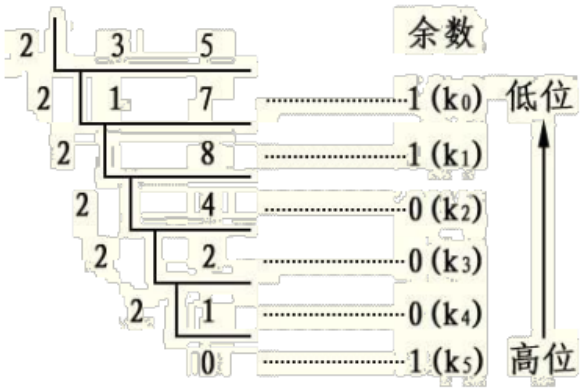


3. 思路拓展

1. 我们会碰到更多种括号：（）、[ ]、{、}
2. 通用括号算法

P18：十进制转为二进制

1. 十进制转换为二进制， 采用的是“除以2求余数”的算法。



2. “除以2”的过程， 得到的余数是从低到高的次序， 而输出则是从高到低， 所以需要一个栈来反转次序。

3. 扩展到更多进制转换

1. 十进制转换为二进制的算法， 很容易可以扩展为转换到任意N进制：

只需要将“除以2求余数”算法改为“除以N求余数”算法就可以。

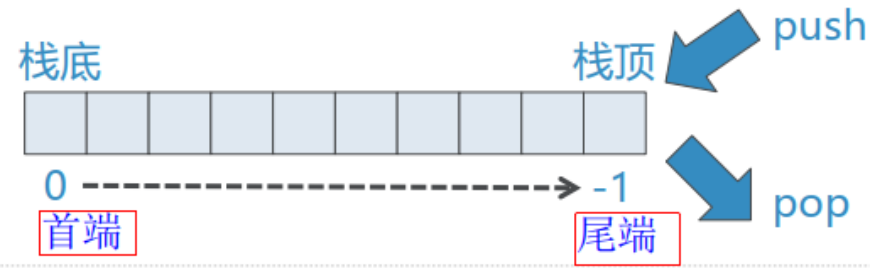
2. 计算机中另外两种常用的进制是八进制和十六进制

P16：？栈的抽象数据类型及其python实现

例子：

3. 文档中的“undo”选项，返回最后操作的一次；
4. 网页中的“Back”选项，返回最后浏览的网页；

2. 定义：



3. 基本操作

❖抽象数据类型“栈”定义为如下的操作

Stack(): 创建一个空栈，不包含任何数据项

push(item): 将item加入栈顶，无返回值

pop(): 将栈顶数据项移除，并返回，栈被修改

peek(): “窥视”栈顶数据项，返回栈顶的数据项但不移除，栈不被修改

isEmpty(): 返回栈是否为空栈

size(): 返回栈中有多少个数据项

4. 操作样例

抽象数据类型Stack：操作样例

Stack Operation	Stack Contents	Return Value
s= Stack()	[]	Stack object
s.isEmpty()	[]	True
s.push(4)	[4]	
s.push('dog')	[4,'dog']	
s.peek()	[4,'dog']	'dog'
s.push(True)	[4,'dog',True]	
s.size()	[4,'dog',True]	3
s.isEmpty()	[4,'dog',True]	False
s.push(8.4)	[4,'dog',True,8.4]	
s.pop()	[4,'dog',True]	8.4
s.pop()	[4,'dog']	True
s.size()	[4,'dog']	2

问题：

？？？如何导入模块Class?

？？？如何在另一平台上导入栈?

P17：？简单括号匹配

？？？力扣题目的实现?

4. 规则：

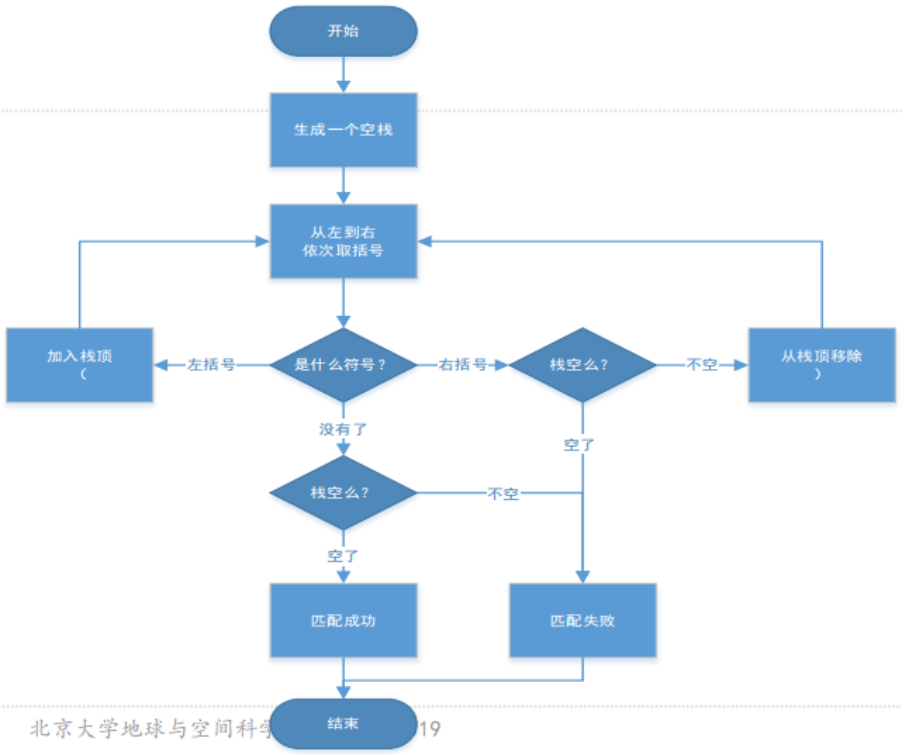
3. 首先，每个开括号要恰好对应一个闭括号；

4. 其次，每对开闭括号要正确的嵌套；

例子：正确：(()()()())， ((((()))))，(()((())()))；

错误：((((((())， ()))， (()()(()；

5. 算法流程图：



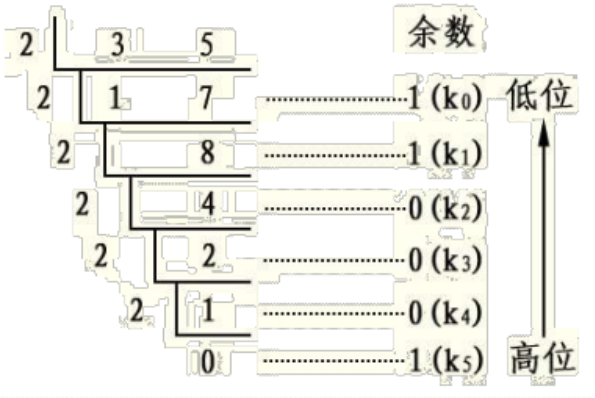
6. 思路拓展

3. 我们会碰到更多种括号：（）、[ ]、{、}

4. 通用括号算法

P18：十进制转为二进制

4. 十进制转换为二进制， 采用的是“除以2求余数”的算法。



5. “除以2”的过程， 得到的余数是从低到高的次序， 而输出则是从高到低， 所以需要一个栈来反转次序。

6. 扩展到更多进制转换

3. 十进制转换为二进制的算法， 很容易可以扩展为转换到任意N进制：只需要将“除以2求余数”算法改为“除以N求余数”算法就可以。

4. 计算机中另外两种常用的进制是八进制和十六进制