

# 實驗21-24

21. 排列

22. 組合

23. 定義class Node，建立運算二元樹

24. 以遞迴程式設計 實作class Node的內建方法  
evalBinaryTree()，回傳二元樹的運算答案

**Ex21**、 $s = \text{"abcde"}$ ，**列舉**  
**所有不同的排列**

Lecture 17

## 定義遞迴函數perm(s)

1. 如果字串s的長度是1，回傳s
2. 令ans為空串列
3. for i in range(len(s)):
  - A. 令t代表剔除字串s第i個字元的字串，求t
  - B. 令items代表字串t的所有排列，以遞迴的方式求items
  - C. for item in items:
    - ① 令p代表s[i]與item合併後的字串
    - ② 將p附加在ans中

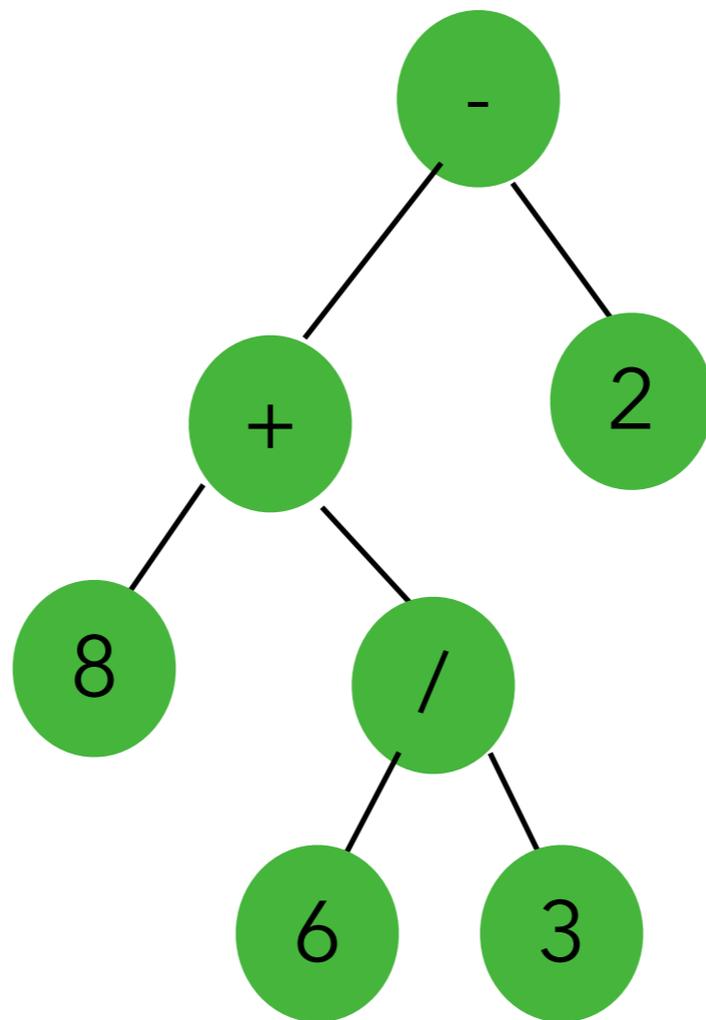
**Ex22**、 $s = "12345"$ ， $r = 2$

**列舉從 $s$ 中任意取 $r$ 個字元  
的所有可能**

## 定義遞迴函數**choose(s,r)**

1. 令**ans**為空串列
2. 如果**r**是1:
  - A. **for i in range(len(s)):**
    - ① 將**s[i]**附加在**ans**
  - B. 回傳**ans**
3. 如果**s**的長度等於**r**:  
將**s**附加在**ans**，回傳**ans**
4. 令**t**為**s[1:]**
5. 以遞迴函數**choose**，列舉在**t**字串中任選**r-1**個字元的所有可能，並儲存在串列**a**中
6. **for item in a:**
  - A. 令**p**代表**s[0]**與**item**合併後的字串
  - B. 將**p**附加在**ans**
7. 以遞迴函數**choose**，列舉在**t**字串中任選**r**個字元的所有可能，並儲存在串列**b**中
8. 回傳 **ans+b**

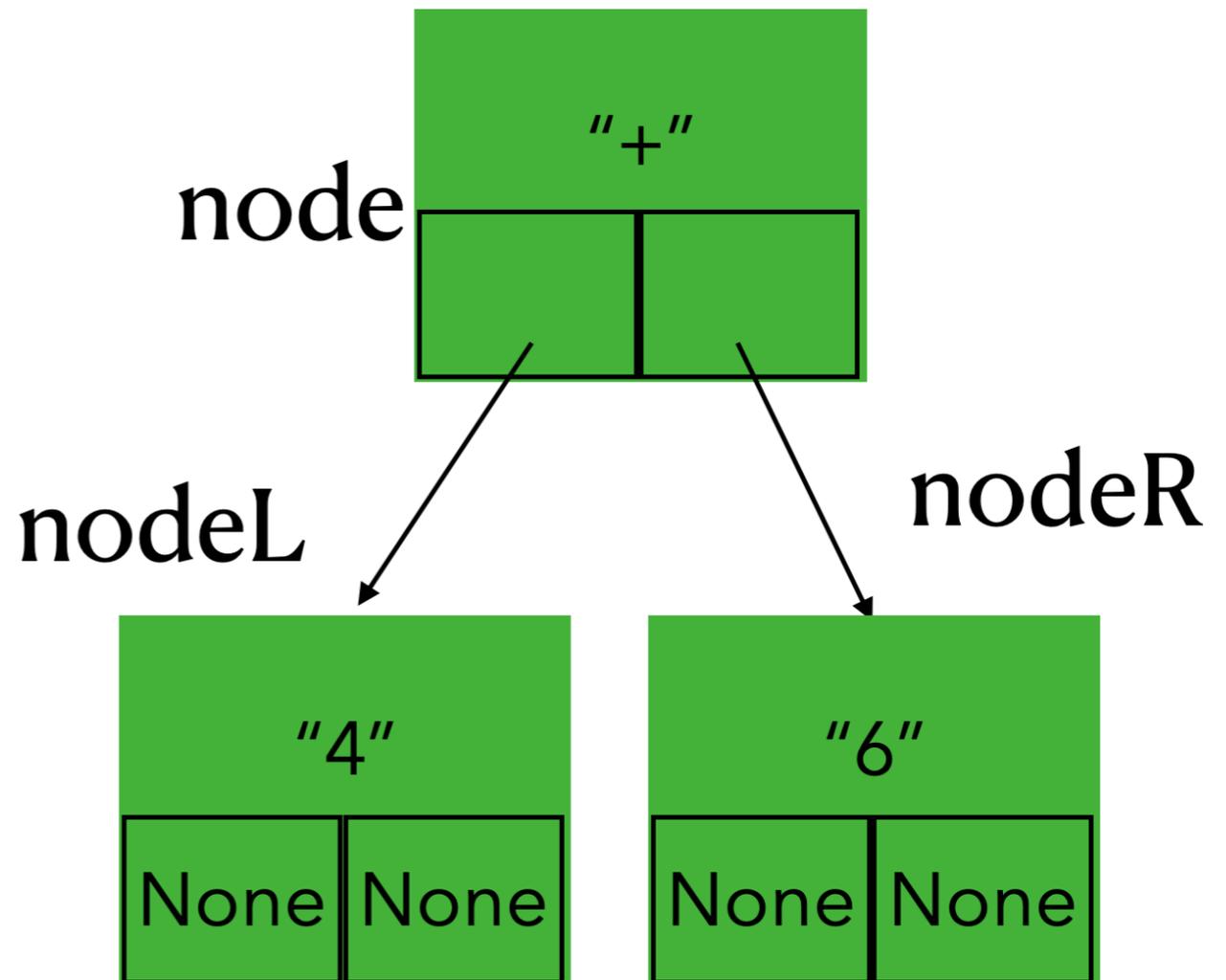
Ex 23. 定義class Node，  
建立運算二元樹



# Step 1. Define class Node

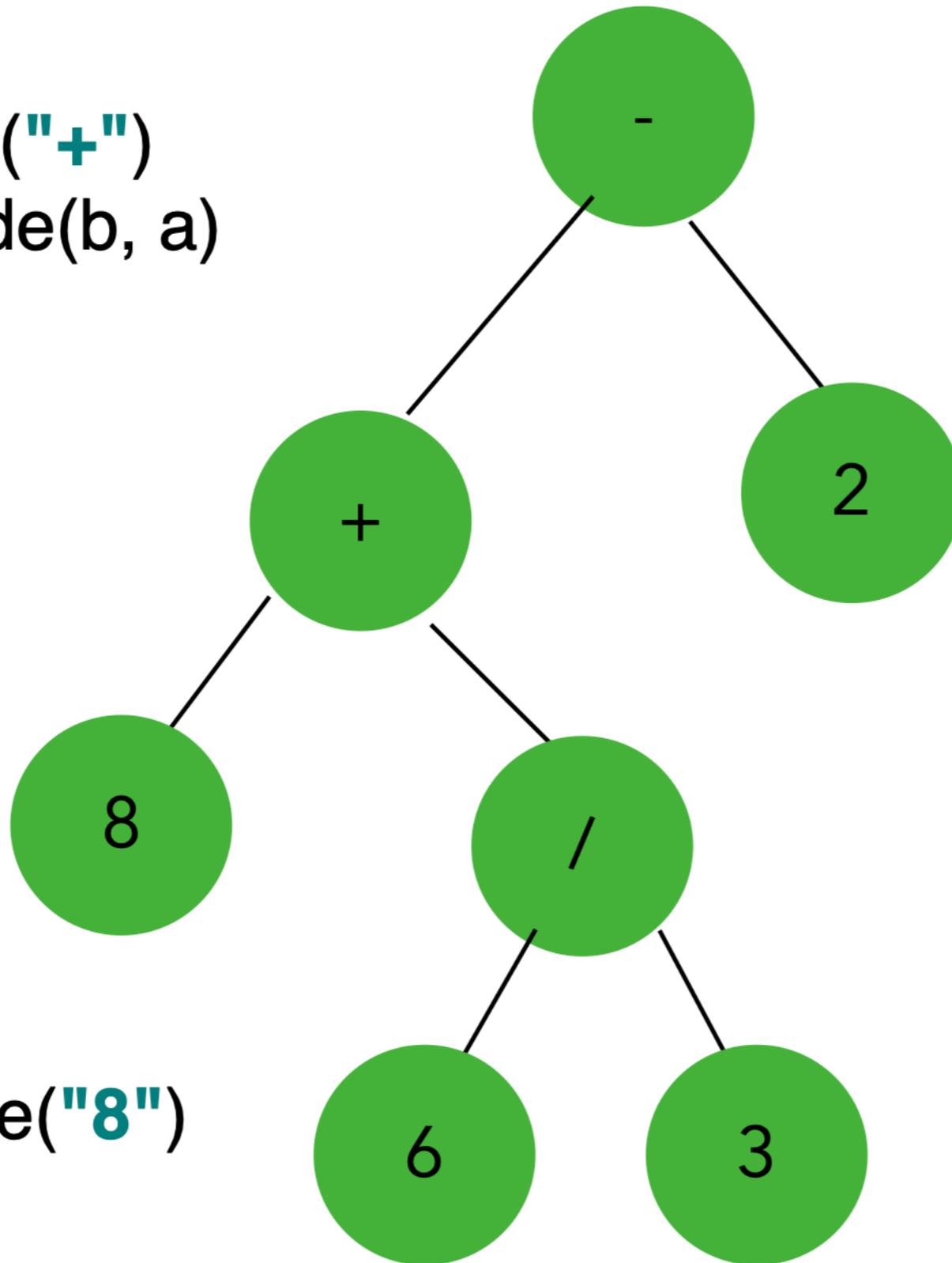
```
class Node:  
    def __init__(self, data):  
        self.data = data  
        self.right = None  
        self.left = None  
  
    def  
    addNode(self, nodeL, nodeR):  
        self.left = nodeL  
        self.right = nodeR
```

```
node = Node("+")  
nodeL = Node("4")  
nodeR = Node("6")  
node.addNode(nodeL, nodeR)
```



**Step 2. Construct a  
binary tree using class  
Node**

```
c = Node("+")  
c.addNode(b, a)
```



```
b = Node("8")
```

```
e = Node("-")  
e.addNode(c,d)
```

```
d = Node("2")
```

```
a = Node("/")  
nl = Node("6")  
nr = Node("3")  
a.addNode(nl,nr)
```

**Ex 20. 以遞迴程式設計實作  
class Node的內建方法  
evalBinaryTree()，回傳二元樹  
的運算答案**

Lecture 18

**Step 1. 定義方法  
evalBinaryTree()**

**Step 2.** ， 檢查節點資料  
是否為數字，如果是數  
字，將該數字字符串回傳

**Step 3.** 如果節點資料不是數字，進行以下步驟：

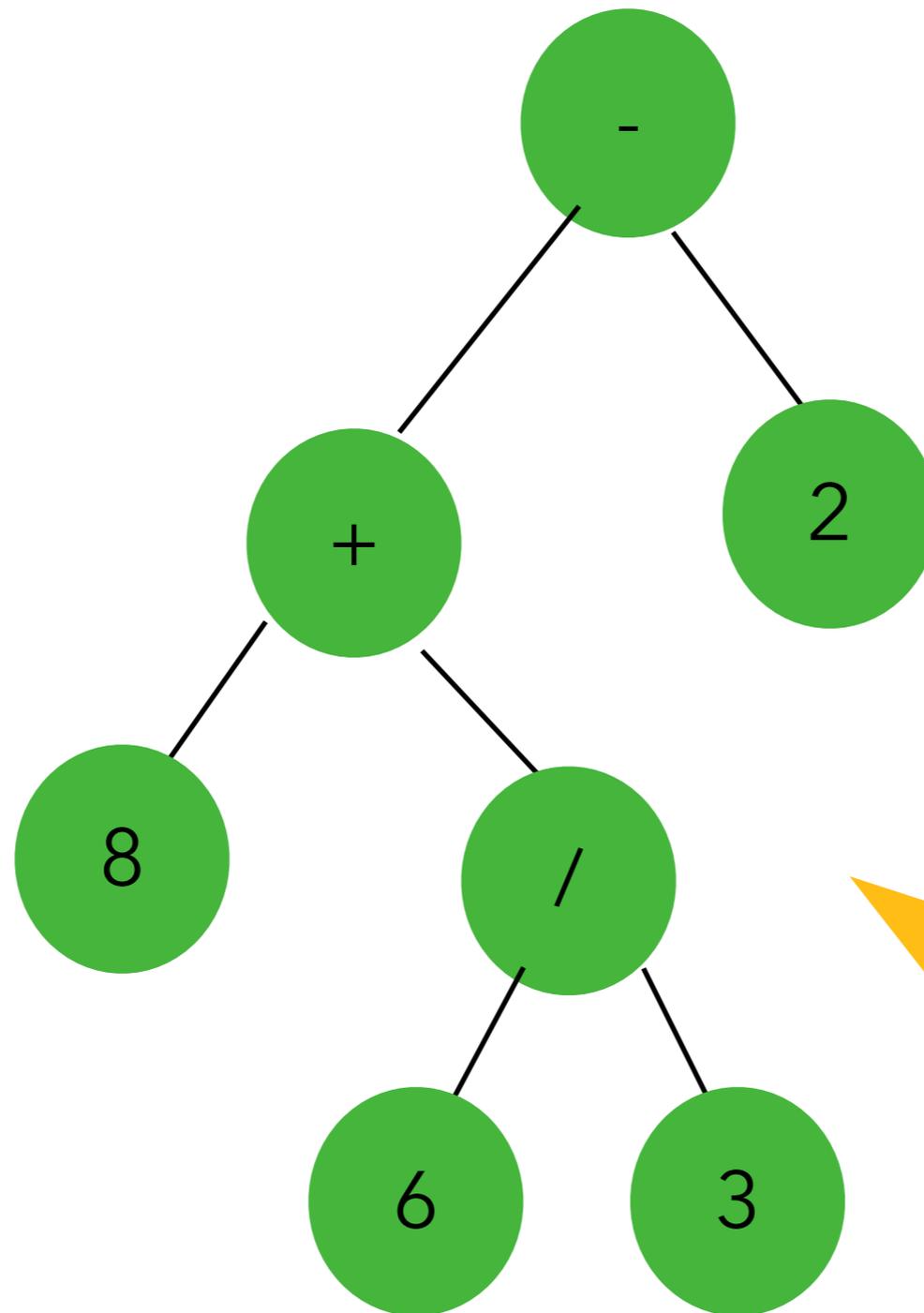
**3A.** 以遞迴方法先求左子樹的運算式答案，儲存在**ansL**中

**3B.** 以遞迴方法先求右子樹的運算式答案，儲存在**ansR**中

**3C.** 應用節點**data**中的運算子，合併**ansL**與**ansR**，算出最後答案，並將答案轉為字串儲存在**ans**中

**3D.** 回傳**ans**

# 應用evalBinaryTree()計算二元樹表示式



Lecture 18