

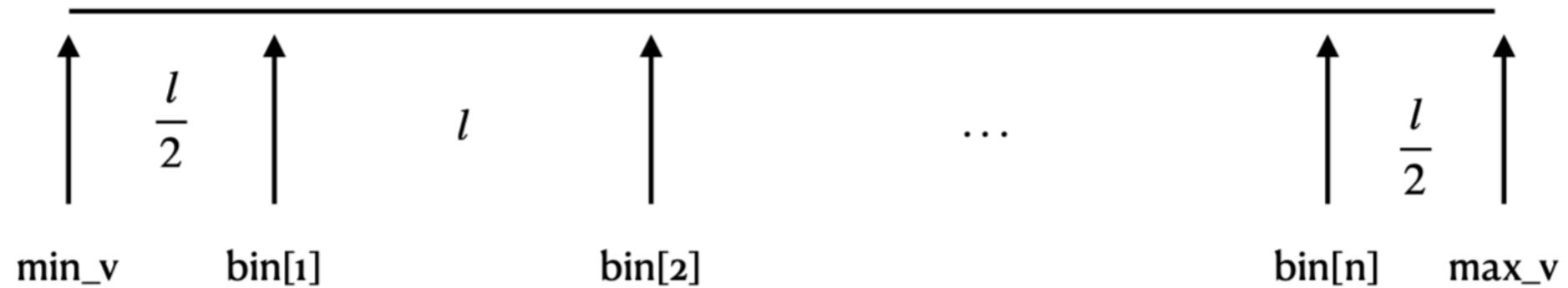
# **2025 Final Review**

$$\text{Interval length } l = \frac{\text{max\_v} - \text{min\_v}}{n}$$

$$\text{bin}[1] = \text{min\_v} + \frac{l}{2}$$

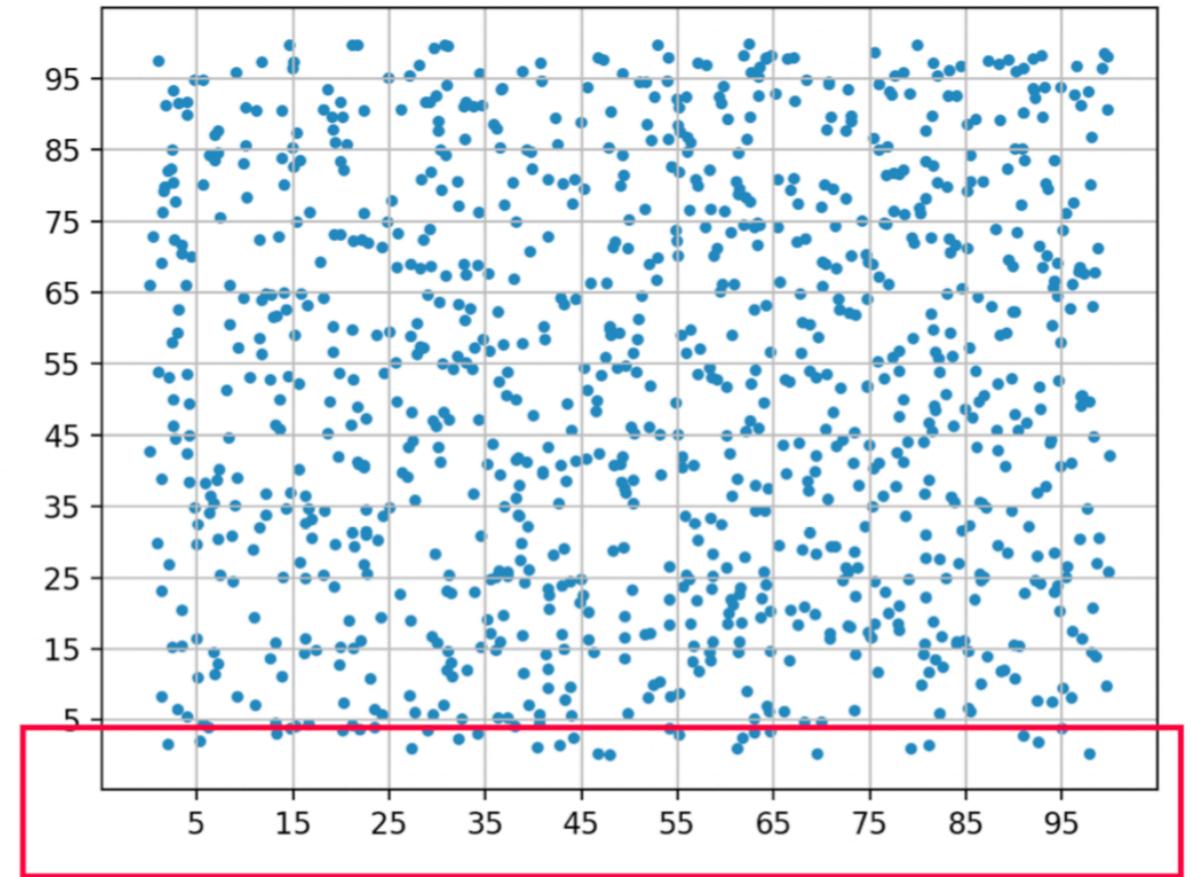
$$\text{bin}[2] = \text{bin}[1] + l \quad \text{bin}[i] = \text{bin}[i-1] + l$$

$$\text{Check point : } \text{bin}[n] = \text{max\_v} - \frac{l}{2}$$



```
def bin_points_list(data_list,n):
    max_v = max(data_list)
    min_v = min(data_list)
    l = (max_v-min_v)/n
    bin = []
    for i in range(n):
        if i == 0:
            bin.append(min_v + l/2)
        else:
            bin.append(bin[i-1]+l)
    return bin
```

令max\_v代表data\_list的最大值，min\_v代表data\_list的最小值，請在區間I = [max\_v,min\_v]取n個點，將區間I等量切割為n個不重疊的子區間，串列bin的元素代表n個切割點



```
def bin_points_list(data_list,n):
```

請完成此程式

```
return bin
```

## 完成my\_histogram函數

```
def my_histogram(x,n): #3B
    bin = bin_points_list(x,n)
    x2bin = np.zeros(n)
    for i in range(len(x)):
        [redacted]
    plt.figure()
    plt.bar(range(n),x2bin)
    return x2bin, bin
```

$$bin[ind] = \min_j \|x[i] - bin[j]\|$$

求ind使得  
bin[ind]成為串列bin的n個元  
素中與x[i]最靠近的值

$$A = \begin{bmatrix} \sin(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) \\ \cos(\frac{\pi}{4}) & -\sin(\frac{\pi}{4}) \end{bmatrix}$$

A 矩陣乘以  
點座標

```
A = np.matrix([[1/np.sqrt(2), 1/np.sqrt(2)], [1/np.sqrt(2), -1/np.sqrt(2)]])
```

## 計算所有 $D_{ij}$ 與 $D$

```
def calculate_D(joint,x2bin,y2bin): # 4
    m = len(x2bin)
    n = len(y2bin)
    N = np.sum(x2bin)
    D = 0
    for i in range(m):
        for j in range(n):
            p = 
            q = 
            if abs(q) > 10**-6:
                D += 
    return D
```

$$\mathbf{D} = \sum_{i,j} D_{ij}$$

$$D_{ij} = q[i,j] \log \frac{q[i,j]}{p[i,j]}$$

期中考加分題(30 points)：將圖一A中的post\_submit.html檔放置在templates目錄，匯入Flask套件後，就可以執行getvalue.py檔。執行後，使用者可以透過網頁的GET模式，輸入字串Jiann-Ming，並在POST模式，看到輸出字串“Hello, Jiann-Ming”。

- A. 請說明你要設計的網頁功能，輸入與輸出間關係為何？
- B. 修改圖一A與圖一B的程式，完成上述問題A的網頁功能。

post\_submit.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Hello Page</title>
</head>
<body>
<form method='post' action="/">
  <input type='text' name='username'>
  <button type='submit'>Submit</
button>
</form>
</body>
</html>
```

圖一A

getvalue.py

```
1  from flask import Flask, request, render_template
2
3  app = Flask(__name__)
4  @app.route("/", methods=['GET', 'POST'])
5
6  def submit():
7      if request.method == 'POST':
8          return 'Hello, ' + request.values.get('username')
9      return render_template('post_submit.html')
10
11  if __name__ == '__main__':
12      app.debug = True
13      app.run()
```

圖一B

```
1 from flask import Flask, request, render_template
2 import sympy as sp
3 from sympy import *
4
5 app = Flask(__name__)
6
7 @app.route("/", methods=['GET', 'POST'])
8 def submit():
9     if request.method == 'POST':
10         x = sp.symbols('x')
11         ss = request.values.get('username')
12         tt = "diff("+ss+")"
13         ans = eval(tt)
14         return '函數'+ss+'的微分: '+str(ans)
15     return render_template('post_submit.html')
16 if __name__ == '__main__':
17     app.debug = True
18     app.run()
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>微分 Page</title>
</head>
<body>
<form method = 'post' action="/">
  <p>使用符號運算sympy進行微分</p>
  <p>輸入函數f(x)</p>
  <p>例如，輸入：x**2+2*x-1，答案：2*x+2</p>
  <p>例如，輸入：cos(x)，答案：-sin(x) </p>
  <input type = 'text' name = 'username' >
  <button type = 'submit'>按鍵傳送</button>
</form>
</body>
</html>
```

$$x^2 - 3x - 4 \rightarrow (x - 4)(x + 1)$$

# HTML設計

## 因式分解

```
>>> ((x**2-2*x+1)).factor()  
(x - 1)**2
```

1.應用sympy撰寫python程式碼評估符號表示式.

A.Expand  $(x + 2y)^2 = ?$  展開表示式.

B.Factorize  $x^3 + 2x^2 - 5x - 6$ . 求表示式的因式分解.

```
import sympy as sp
x = symbols('x')
y = symbols('y')
```

```
((x + 2*y)**2).expand()
```

```
(x**3 + 2*x**2 - 5*x - 6).factor()
```

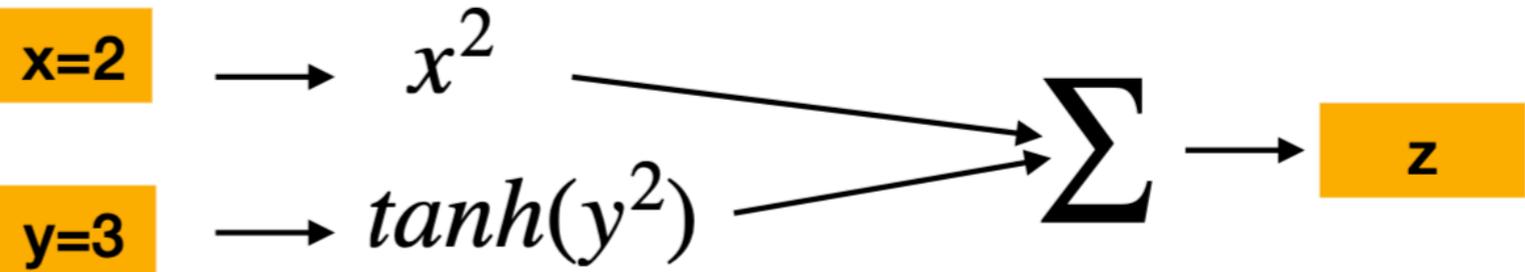
## 請解釋程式碼

```
x = tf.Variable(2.0)
y = tf.Variable(3.0)
```

```
with tf.GradientTape() as t:
    y_sq = tf.math.tanh(y**2)
    z = x**2 + y_sq
```

```
grad = t.gradient(z, {'x': x, 'y': y})
```

```
print('dz/dx:', grad['x']) # 2*x => 4
print('dz/dy:', grad['y'])
```



1. 請以遞迴程式設計解 $n$ 個不同字元的排列問題.

A. 請定義函數，包含函數名稱與輸入參數.

B. 請描述定義本題為原始基本問題的條件.

C. 請寫下解決原始基本問題的程式碼.

D. 請寫下解決本題一般問題的遞迴程式碼.

2.

## 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
4. 回傳ans

1. 以遞迴程式設計解從 $n$ 個不字元選 $r$ 個字元的組合問題，其中 $r \leq n$ .
  - A. 請定義函數，包含函數名稱與輸入參數.
  - B. 請描述定義本題為原始基本問題的條件.
  - C. 請寫下解決原始基本問題的程式碼.
  - D. 請寫下解決本題一般問題的遞迴程式碼.

## 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**

btree.py

(20 points) Given btree.py, please write python codes for constructing binary tree.

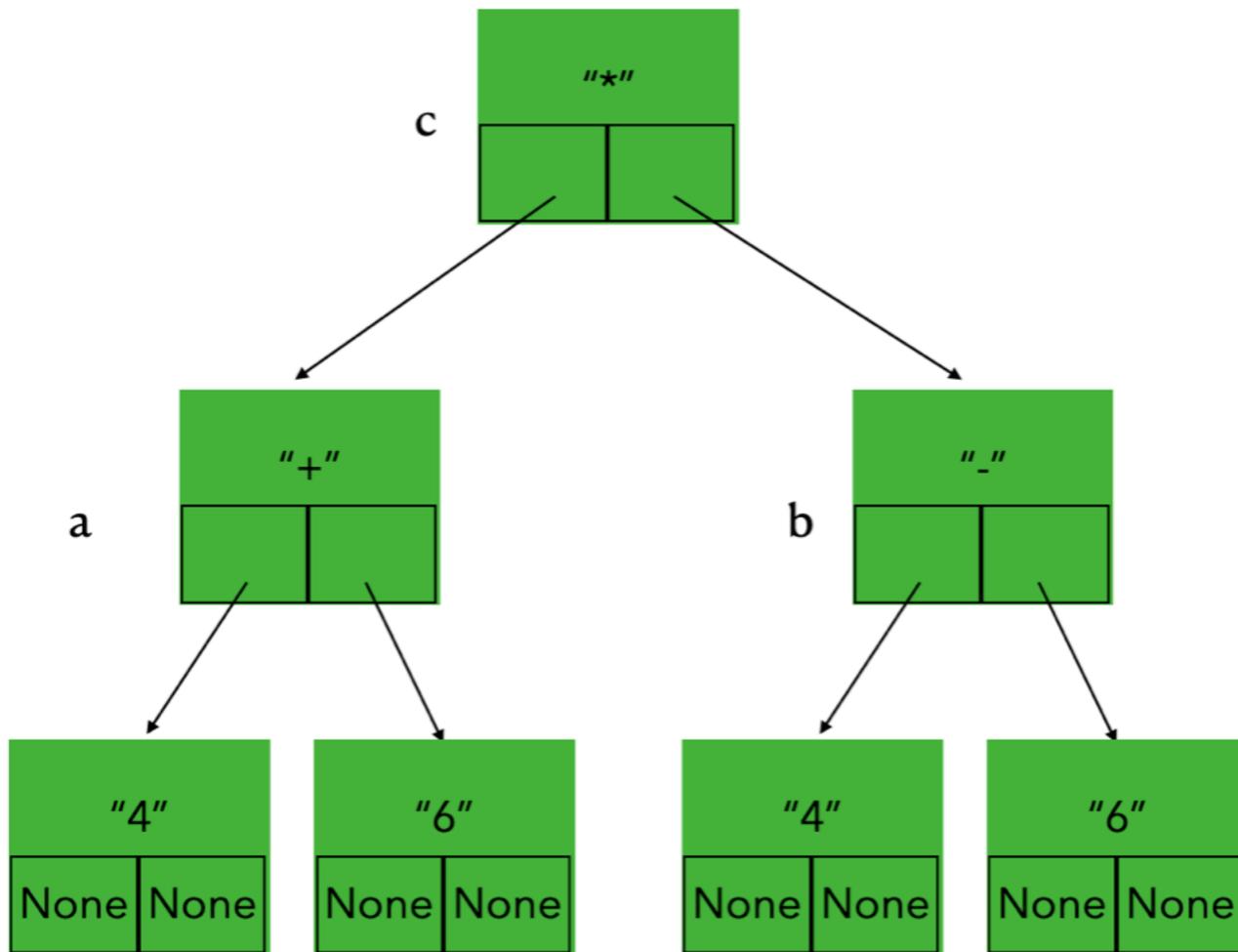
給定btree.py，寫python程式以建構二元樹。

- A. Write codes to construct the binary tree in figure 1. 寫python程式以建構圖1的二元樹
- B. Write codes to construct the binary tree in figure 2. 寫python程式以建構圖2的二元樹.

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

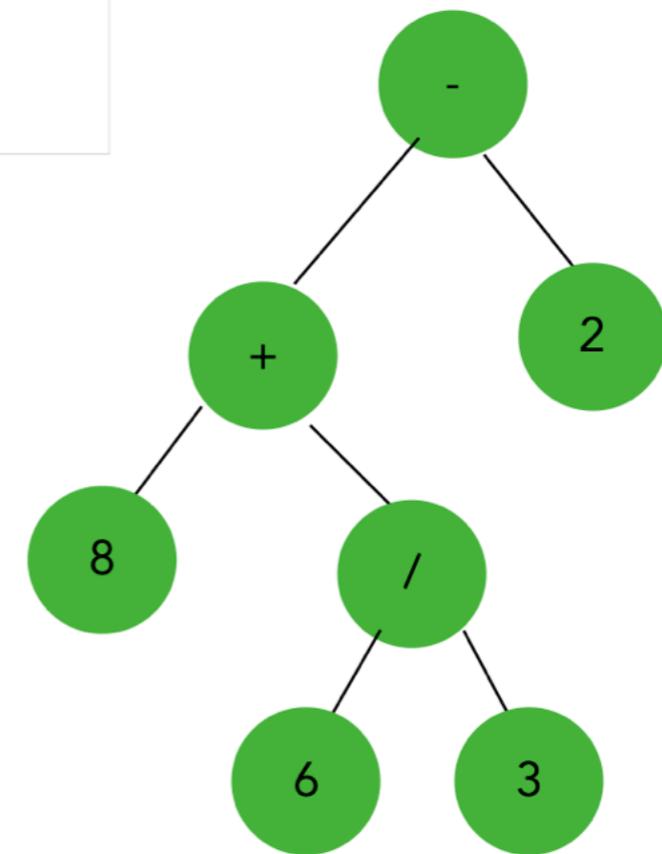
    def addNode(self,nodeL,nodeR):
        self.left = nodeL
        self.right = nodeR
```

Figure 1



```
n = Node("+")
nl = Node("4")
nr = Node("6")
n.add(nl,nr)
```

Figure 2



A. 計算的二元運算樹·相應的遞迴問題的基本問題形式是什麼？又該如何將原問題分解為降階後的子問題，分別以遞迴的方式求取子問題的答案，並透過進一步的合成，求得原問題的答案。

(20 points) 令A為 $n \times n$ 的矩陣， $n$ 為正整數，請回答以下問題，說明如何透過遞迴方法，完成圖九的程式碼，求A的行列式。

- A. 此遞迴問題的基本問題形式是什麼？
- B. 圖九中相對應於上述問題A的程式碼為何？
- C. 如何將A的行列式問題分解為降階後的子問題，分別以遞迴的方式求取子問題的答案，再透過合成，求得原問題的答案。
- D. 圖九中相對應於上述問題C的程式碼為何？

```
def mydet(A):
```

```
    n,m = A.shape
```

```
    if n == 2 and m == 2:
```

```
        [REDACTED]
```

```
    else:
```

```
        [REDACTED]
```

```
    return ans
```

圖九

```
def mydet(A):  
    n,m = A.shape  
    if n == 2 and m == 2:  
        return [redacted]  
    else:  
        ans = 0  
        for i in range(0,n):  
            [redacted]  
        return ans
```

B 為 $(n-1) \times (n-1)$   
矩陣

問題大小 $n$ 不是初始問題，使用前一階問題的答案，合成問題大小 $n$ 的答案表示式