

第2次練習-練習-PC2

學號：112111215

姓名：莊博勳

作業撰寫時間：180 (mins · 包含程式撰寫時間)

最後撰寫文件日期：2024/12/29

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

- ☒ 說明內容
- ☒ 個人認為完成作業須具備觀念

1. 請參照Topic2投影片p. 39，撰寫一隻1.py的程式用於儲存多項式方式儲存多項式

Ans:

```
class Term:
    def __init__(self, coefficient, exponent):
        self.coefficient = coefficient
        self.exponent = exponent

    def evaluate(self, x):
        return self.coefficient * (x ** self.exponent)

class Polynomial:
    def __init__(self):
        self.terms = []

    def add_term(self, coefficient, exponent):
        self.terms.append(Term(coefficient, exponent))

    def evaluate(self, x):
        return sum(term.evaluate(x) for term in self.terms)

# 建立多項式
poly = Polynomial()
poly.add_term(6, 2)
poly.add_term(4, 1)
poly.add_term(2, 0)

# 計算結果
x = 91
result = poly.evaluate(x)
print(f"f({x}) = {result}")
```

2. 承1，請使用物件導向方式實作上題，也就是每個單位的x次方做成一個類別後，完成上述儲存功能，並算其結果。

Ans:

```
class Term:
    """
    表示多項式中的一個項次，例如  $6x^2$ 。
    """
    def __init__(self, coefficient, exponent):
        self.coefficient = coefficient # 係數
        self.exponent = exponent      # 次方

    def evaluate(self, x):
        """
        計算該項次在指定 x 值的結果。
        """
        return self.coefficient * (x ** self.exponent)

    def __str__(self):
        """
        轉換為可讀的多項式項次表示。
        """
        if self.exponent == 0:
            return f"{self.coefficient}"
        elif self.exponent == 1:
            return f"{self.coefficient}x"
        else:
            return f"{self.coefficient}x^{self.exponent}"

class Polynomial:
    """
    表示整個多項式，由多個項次組成。
    """
    def __init__(self):
        self.terms = [] # 儲存多項式的項次

    def add_term(self, coefficient, exponent):
        """
        添加一個項次到多項式中。
        """
        self.terms.append(Term(coefficient, exponent))

    def evaluate(self, x):
        """
        計算整個多項式在指定 x 值的結果。
        """
        return sum(term.evaluate(x) for term in self.terms)

    def __str__(self):
        """
        將整個多項式轉換為可讀表示。
        """
        if not self.terms:
            return "0"
```

```

        return " + ".join(str(term) for term in self.terms)

# 建立多項式
poly = Polynomial()
poly.add_term(6, 2) # 6x^2
poly.add_term(4, 1) # 4x
poly.add_term(2, 0) # 2

# 輸出多項式表示
print("Polynomial:", poly)

# 計算結果
x = 91
result = poly.evaluate(x)
print(f"f({x}) = {result}")

```

3.

Ans:

```

class SparseMatrix:
    def __init__(self):
        self.data = {}

    def gray(self, i, j, value):
        if value != 0:
            self.data[(i, j)] = value

    def display(self):
        for (i, j), value in self.data.items():
            print(f"Pixel ({i}, {j}) = {value}")

# 建立稀疏矩陣
matrix = SparseMatrix()
matrix.gray(0, 1, 50)
matrix.gray(1, 3, 120)
matrix.gray(2, 4, 180)
matrix.gray(3, 2, 255)

# 顯示結果
matrix.display()

```

4.

Ans:

```

def count_inversions(arr):
    """

```

計算陣列中的倒轉成對數量。

倒轉成對的定義是對於索引 i 和 j ，若 $i < j$ 且 $A[i] > A[j]$ ，則 (i, j) 是一個倒轉成對。

```
"""
count = 0
n = len(arr)
for i in range(n):
    for j in range(i + 1, n):
        if arr[i] > arr[j]:
            count += 1
return count

# 主程式
if __name__ == "__main__":
    # 輸入陣列大小與元素
    n = int(input("Enter the number of elements in the array: "))
    arr = list(map(int, input("Enter the elements of the array separated by
spaces: ").split()))

    # 確保輸入的陣列長度正確
    if len(arr) != n:
        print("Error: The number of elements does not match the specified size.")
    else:
        # 計算倒轉成對數量
        inversions = count_inversions(arr)
        print(f"Number of inversions: {inversions}")
```

個人認為完成作業須具備觀念

開始寫說明，需要說明本次練習需學會那些觀念 (需寫成文章，需最少50字，並且文內不得有你、我、他三種文字)且必須提供完整與練習相關過程的notion筆記連結