資料結構

HW2 Problem1 & Problem2

日期:2024/11/27

CHAPER1:解題說明

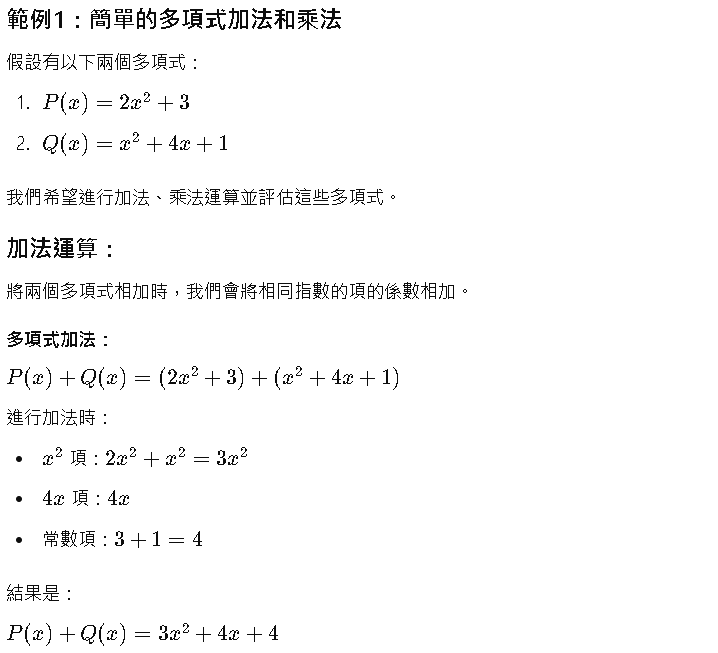
**想法 (How to do?)**

本題的目的是實現多項式的加法、乘法和評估，並進行正確的運算。多項式通常表達為各個項的係數和指數的組合，像是：

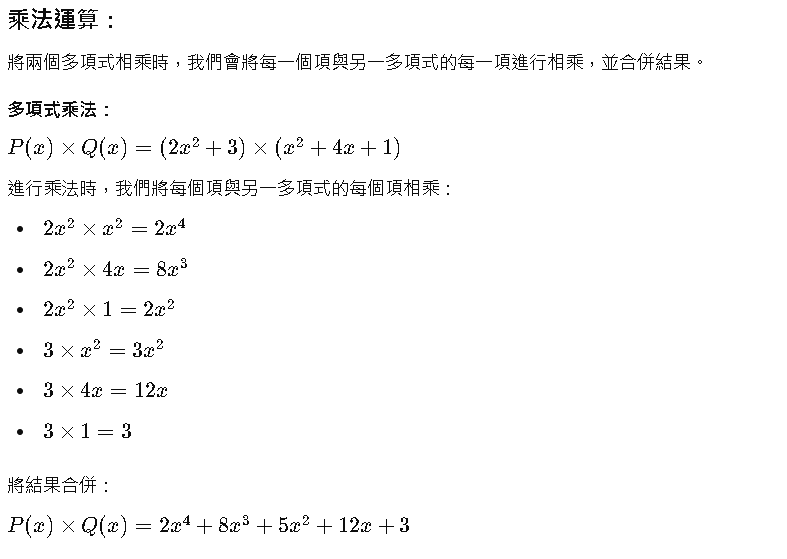
* 加法：將相同指數的項的係數相加。
* 乘法：將每個項的係數相乘，指數相加。
* 評估：將多項式代入某個數值，計算結果

**存取和比對多項式的步驟：**

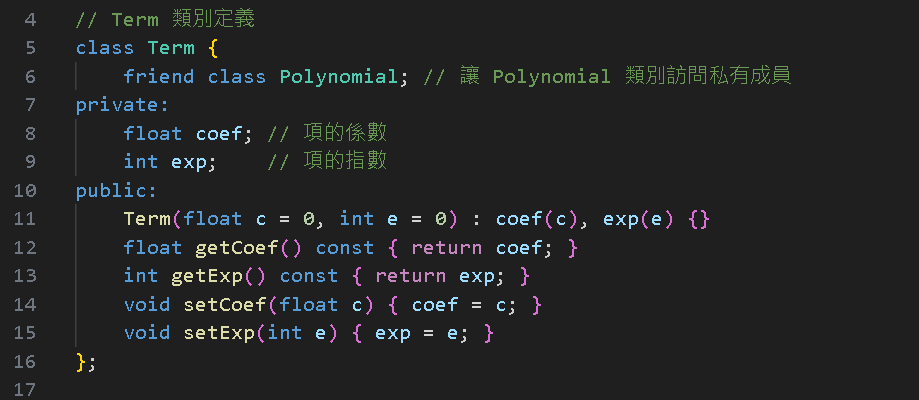
1. **比較運算符**：
   * 我們需要定義一個運算符重載，以便在進行加法和乘法運算時，可以比較兩個項的指數。
   * 比較的規則是：**指數較大的項優先**，即按指數從高到低排序。
2. **存取多項式項**：
   * 使用 Term 類別來表示單一的多項式項，並存取每個項的係數和指數。
   * 透過一個數組或動態數組來存儲所有項。
3. **加法和乘法的比對邏輯**：
   * 在加法中，我們需要遍歷兩個多項式，對每個項的指數進行比較，並合併係數相同的項。
   * 在乘法中，則是對每個項與每個其他項進行比較，並根據規則相加指數，並存儲結果

****

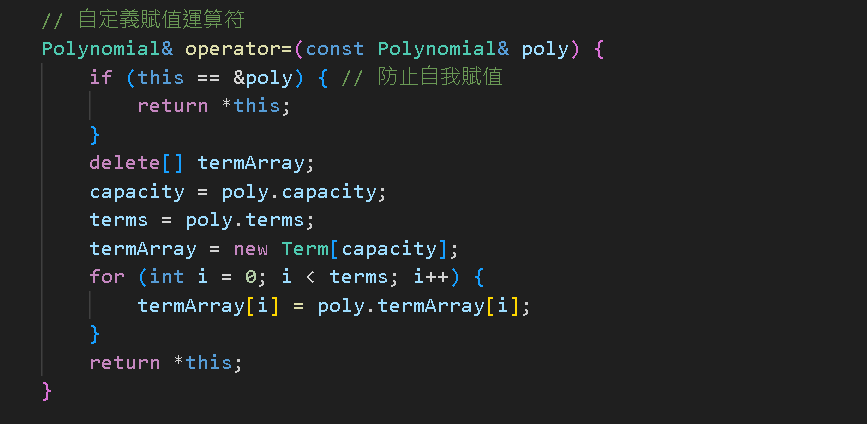


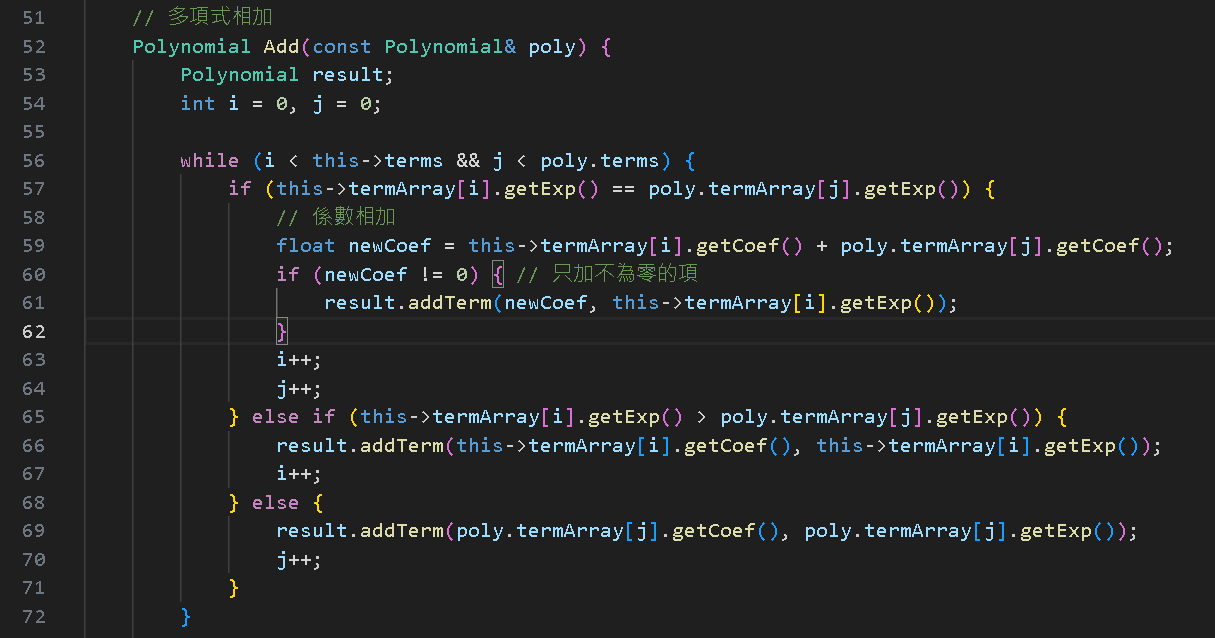


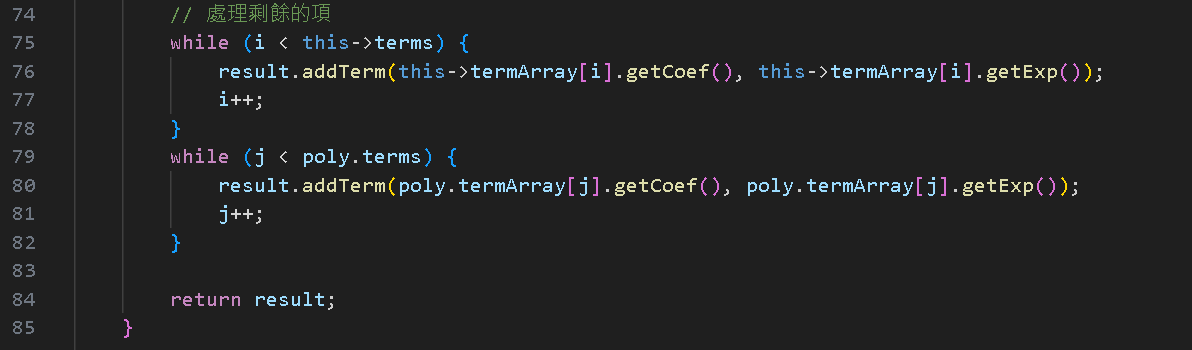
CHAPER2: Algorithm Design & Programming

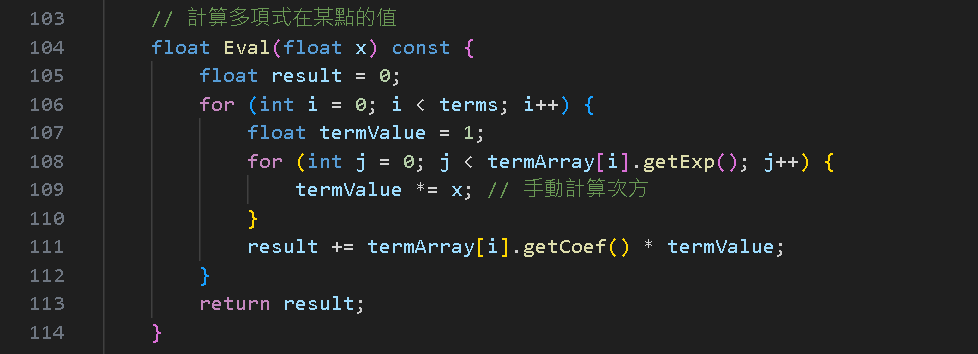
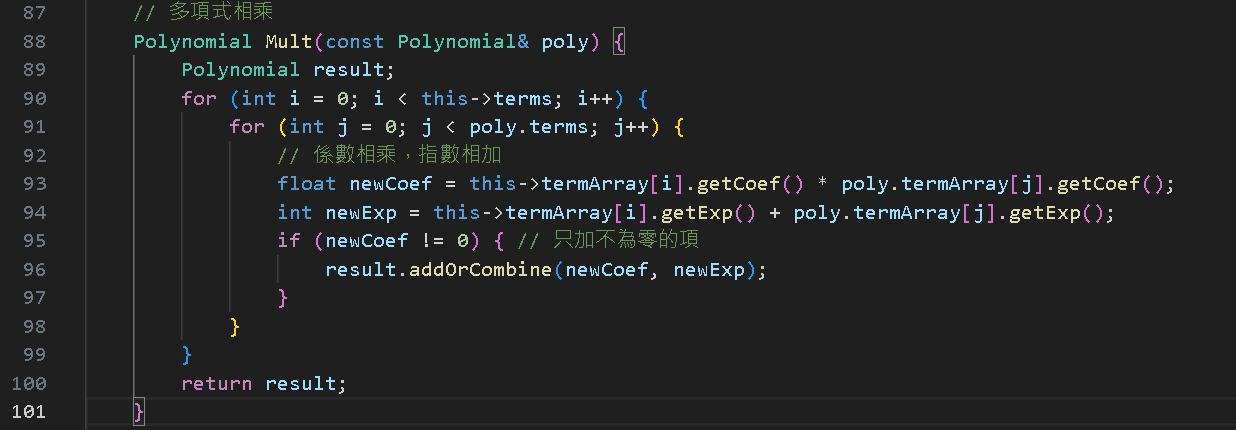


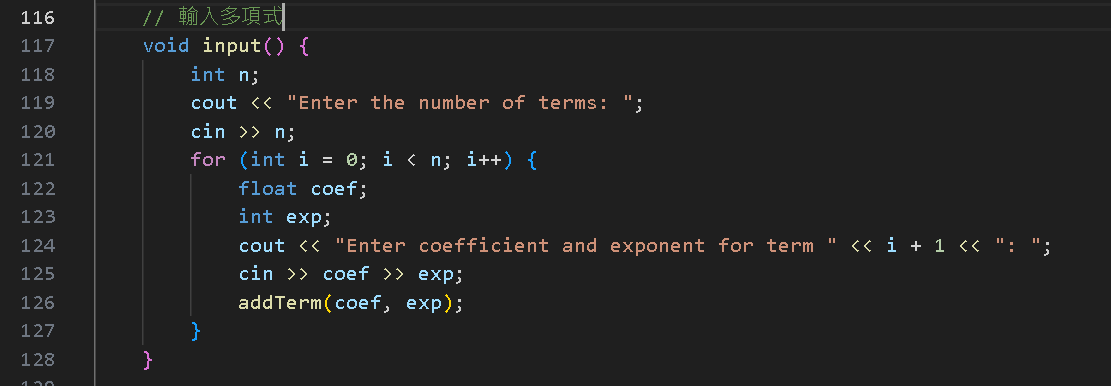


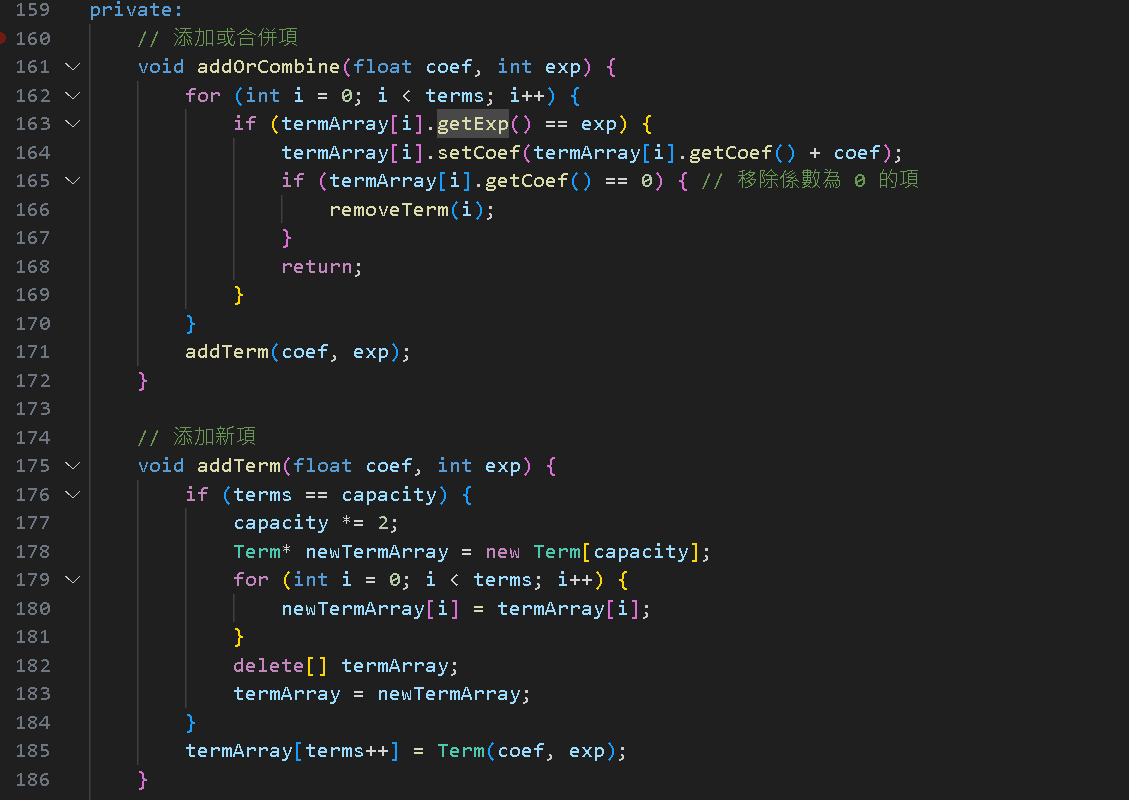


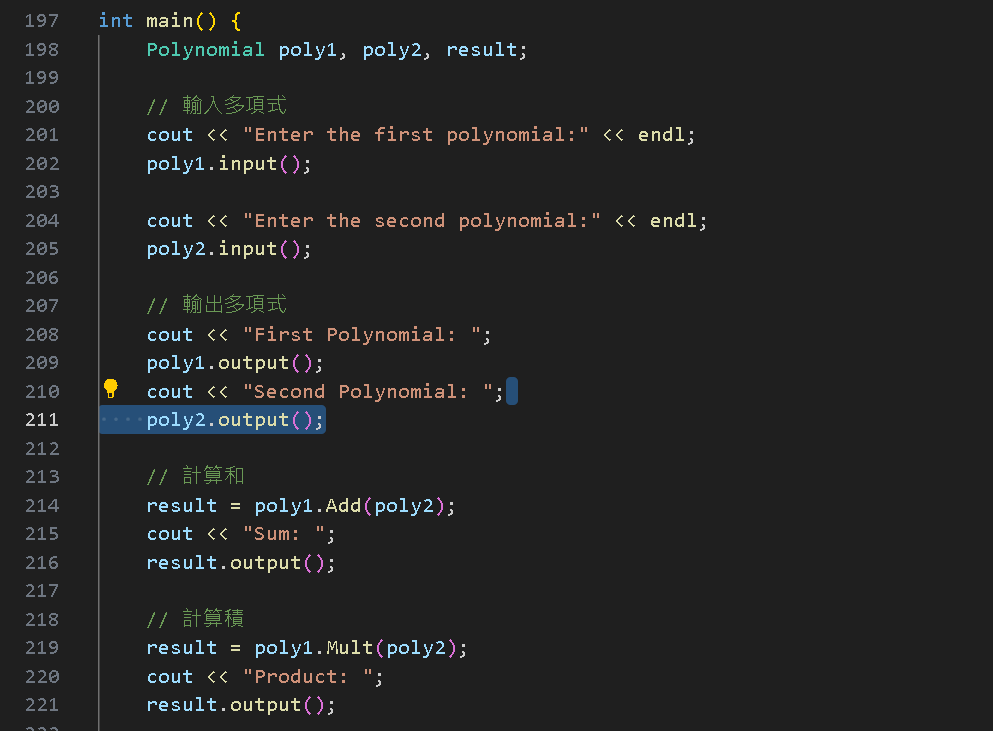
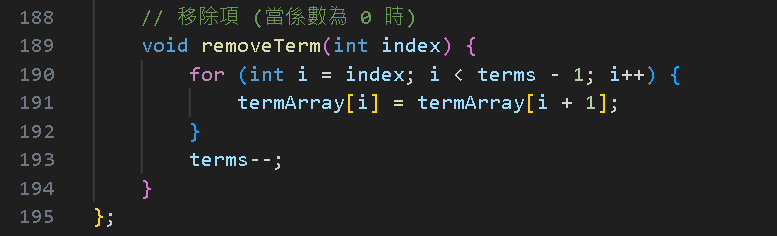


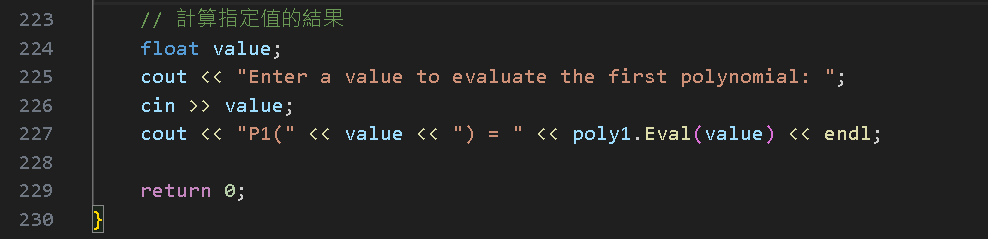












CHAPER3:效能分析

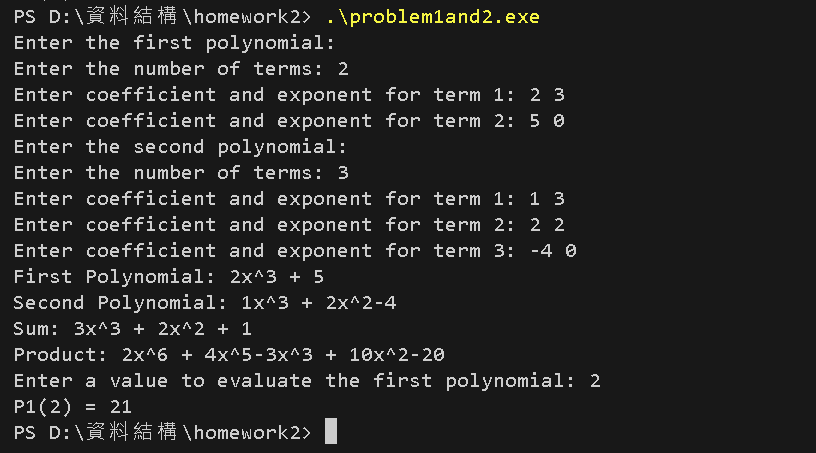
**Time Complexity:**

* **加法 (Add)**: 每個多項式的項數是 n 和 m，因此加法的時間複雜度是 O(n+m)O(n + m)O(n+m)，其中 n 和 m 分別是兩個多項式的項數。
* **乘法 (Mult)**: 每個多項式的項數是 n 和 m，因此乘法的時間複雜度是 O(n×m)O(n \times m)O(n×m)，其中 n 和 m 是兩個多項式的項數。
* **評估 (Eval)**: 評估多項式的時間複雜度是 O(n)O(n)O(n)，其中 n 是多項式的項數，因為每項需要計算其次方。

**Space Complexity:**

* **空間複雜度**：空間複雜度主要取決於多項式的項數，最壞情況下是 O(n+m)O(n + m)O(n+m) 空間，其中 n 和 m 分別是兩個多項式的項數。

CHAPER4:測試與認證



CHAPER5:效能量測

可根據項數量級測量效能。例如，在多項式的項數量為 1000 和 10000 時進行性能測量，並測量加法、乘法和評估的執行時間。

CHAPER6:心得討論

這次的作業讓我深入了解多項式的運算，雖然這個作業的設計主要是基於加法和乘法的基本運算，但在實際應用中，我學到了如何處理多項式中可能出現的各種邊界情況。例如，處理某些項的係數為零的情況，避免它們在最終結果中出現。

將每個功能模塊（加法、乘法、評估等）封裝為獨立的成員函數，這樣不僅讓程式結構清晰，還能夠讓其他人容易理解和使用。

在這次作業中，我使用了 Term 類別來表示多項式的單個項，這讓我能夠將每個項的係數和指數封裝成對象。這樣的設計讓處理和儲存多項式變得更加直觀，並且使後續的加法、乘法等運算變得簡單可控。