

資料結構作業二報告

解題說明

本題目要求設計一個多項式類別 (Polynomial)，並實現以下功能：

1. 多項式輸入與輸出：
 - 使用者可以以人類可讀的格式輸入多項式，程式將解析輸入並建構對應的物件。
 - 支援以標準輸出顯示多項式內容。
2. 多項式基本運算：
 - 實現多項式的加法 (Add) 和乘法 (Mult)。
3. 多項式值的計算：
 - 提供一個函數計算多項式在特定自變數 (x) 值下的結果 (Eval)。
4. 效能與使用性：
 - 優化記憶體使用，並處理不同輸入格式，確保程式穩定運作。

程式實作

主要設計重點

1. 使用類別 Polynomial 來封裝多項式的數據和操作行為，將多項式儲存為一組包含係數和指數的陣列。
2. 使用動態記憶體管理 (new 和 delete) 來管理多項式的儲存空間，確保靈活性。
3. 支援使用者以類似數學表達式的格式輸入多項式，並以適當格式輸出。

效能分析

時間複雜度

1. 多項式加法：
 - 時間複雜度為 $O(n)O(n)O(n)$ ，其中 nnn 是最大指數值（取決於最高次多項式）。
2. 多項式乘法：
 - 時間複雜度為 $O(n^2)O(n^2)O(n^2)$ ，因為對於每個多項式的每一項，需要與另一個多項式的每一項相乘。
3. 多項式計算值：
 - 時間複雜度為 $O(n)O(n)O(n)$ 。

空間複雜度

1. 動態記憶體使用量與多項式最高次數相關，為 $O(n)O(n)O(n)$ 。
2. 輸入與輸出暫存空間的需求較小，影響可忽略。

測試與驗證

測試案例

1. 基本功能測試：
 - 輸入多項式： $3x^2 + 1$ 和 $x + 1$
 - 測試加法與乘法運算。(執行圖在.png 檔)
2. 邊界條件測試：
 - 輸入零多項式或單項式，如 0 或 x 。
 - 測試負數係數與高次多項式。
3. 效能測試：
 - 輸入高次多項式（例如，**1000** 項）以測試記憶體管理與運算效能。

驗證結果

- 輸出格式正確。
- 運算結果與理論值一致。
- 高次多項式的運算時間可接受，無記憶體洩漏。

申論及開發報告

功能總結

該程式成功實現了多項式的建構、運算與求值功能，具有良好的使用性。

開發難點

- 輸入解析需要處理多種格式，並確保兼容性。
- 動態記憶體管理需小心處理，以避免洩漏與錯誤。

未來改進方向

- 提高效能：
 - 將 Mult 優化為稀疏多項式的方式，減少不必要的零項運算。
- 擴展功能：
 - 添加微分與積分功能。
 - 支援多項式除法。
- 友好介面：
 - 使用圖形化界面或更直觀的交互方式提升使用體驗。

效能總結

在合理的輸入範圍內，程式能高效運行，並提供正確結果。