

書面報告

DEMO: <https://youtu.be/nWG4eKSIQNg>

簡要說明：

本次實作多項式是以 `array` 實作，另外多項式的係數是浮點數型別，次方設定為整數型別，限制最多創建 10 個多項式，且總非零項數目不超過 100 項。

輸出入介面說明：

初始選擇功能的畫面，依照功能輸入，需先創立多項式，才能執行某些功能

Options:

c	Create new polynomials
a	Add specific term to a specific polynomial
d	Delete specific term from a specific polynomial
p	Print specific polynomial
s	Show specific term of a polynomial
A	Arithmetic Calculation
r	Stop

Your Option:

功能介紹：

(1) 創建多項式(Create new polynomials)

在創建多項式時，會先要求輸入要創建多少個多項式，再一一詢問多項式名字、非零項數目、係數與次方。

Function :CreatePoly()

令全部多項式非零項數總和為 n

$O(\text{全部多項式非零項數總和}) = O(n)$

注意：

1. 創建多項式時，需一次創建多個，無法分批創建。
2. 輸入多項式時，請根據次方大小，由最大次方依序輸入至最小次方
3. 名字長度最多為 4，無法創建相同名字的多項式

(2) 新增多項式 P 的非零項 (Add specific term to a specific polynomial)

在新增非零項前，會先詢問多項式 P 的名字，在詢問要新增的係數與次方。
此功能可以新增已經存在的次方或全新的次方。

Function: Add_Del_Term(int act)

流程：

1. 尋找多項式 P
2. 尋找新增值欲加入的 index
3. 往後搬移 index 後的非零項
4. 將新增值加入多項式 P
5. 更新所有多項式的 start index 與 end index

時間複雜度：

令多項式總數為 n，多項式 P 的非零項數為 m、index 值後的非零項數為 k
 $O(\text{多項式總數} + \text{多項式 P 的非零項數} + \text{index 值後的非零項數} + \text{多項式總數}) = O(2n+m+k)$

(3) 刪除多項式 P 的非零項 (Delete specific term from a specific polynomial)

在刪除非零項前，會先詢問多項式 P 的名字，在詢問要刪除的次方。

Function: Add_Del_Term(int act)

流程：

1. 尋找多項式 P
2. 尋找欲刪除次方的 index
3. 往前搬移 index 後的非零項（刪除欲刪除得值）
4. 更新所有多項式的 start index 與 end index

時間複雜度：

令多項式總數為 n，多項式 P 的非零項數為 m、index 值後的非零項數為 k
 $O(\text{多項式總數} + \text{多項式 P 的非零項數} + \text{index 值後的非零項數} + \text{多項式總數}) = O(2n+m+k)$

(4) 顯示多項式 P 的內容(Print specific polynomial)

在顯示之前會先詢問多項式 P 的名字，再印出多項式 P。

Function: Print()

流程：

1. 尋找多項式 P
2. 顯示多項式 P

時間複雜度：

令多項式總數為 n ，多項式 P 的非零項數為 m
 $O(\text{多項式總數} + \text{多項式 P 的非零項數}) = O(n+m)$

(5) 指定多項式 P 之指數次方，顯示該項係數值(Show specific term of a polynomial)

在顯示之前會先詢問多項式 P 的名字與次方數，再印出該項係數。

Function: ShowCoef()

流程：

1. 尋找多項式 P
2. 尋找欲顯示的次方

時間複雜度：

令多項式總數為 n ，多項式 P 的非零項數為 m
 $O(\text{多項式總數} + \text{多項式 P 的非零項數}) = O(n+m)$

(6) 算術運算(Arithmetic Calculation)

運算式子為 $P2_P1$ ，會依序詢問 P2 與 P1 多項式的名字，在詢問要執行(+、-、*)等運算。

Function: Arithmetic()

流程：

1. 尋找多項式 P2
2. 尋找多項式 P1

時間複雜度：

令多項式總數為 n
 $O(\text{多項式總數} * 2) = O(2*n) = O(n)$

執行加與減：

Function: Plus_Minus(int index_p2, int index_p1, char ao)

流程：

1. 比較 P2 P1 係數
2. 做相對應處理
3. 印出答案

時間複雜度：

令 m、n 分別為 P2 與 P1 中非零項數目，worst case 是 P2 P1 沒有相同的次方數，假設 P2 會先被掃完

$$O(m+n+m+n) = O(2m+2n) = O(m+n)$$

執行乘：

Function: Mutiplication(int index_p2, index_p1)

流程：

1. P2 係數依序乘以 P1 係數
2. 看次方有沒有出現過
3. 根據次方排序
4. 印出答案

時間複雜度：

令 m、n 分別為 P2 與 P1 中非零項數，worst case 是乘完後次方數沒有相同(在找是否有重複次方時，沒找到)

$$O(m*n + (m*n-1)(m*n-2)/2 + (m*n-1)(m*n-2)/2 + m*n) = O(m^2 * n^2)$$

(7) 停止(Stop)

輸入 Option 時，請輸入 r