資料結構—HW2

作者:

黃昱翔

學號

40932626

December 1, 2024

Contents

Chapter 1

解題說明

實作一個多項式類別，並支援多項式的基本操作（加法、乘法、評估等）。考慮將每一個項（包含係數與次數）設為一個結構或類別，並使用動態陣列儲存非零項。核心在於動態記憶體管理與多項式運算的正確實作

Chapter 2

程式實作

```
class Polynomial {
public:
    Polynomial(int capacity = 10); // 建構式
    ~Polynomial();                 // 解構式

    Polynomial Add(const Polynomial& poly) const; // 加法
    Polynomial Mult(const Polynomial& poly) const; // 乘法
    float Eval(float x) const;                    // 評估多項式

    friend ostream& operator<<(ostream& os, const Polynomial& poly); // 輸出運算子
    friend istream& operator>>(istream& is, Polynomial& poly);       // 輸入運算子

private:
    Term* termArray; // 非零項的陣列
    int capacity;    // 陣列容量
    int terms;       // 非零項數量
    void Expand();   // 動態擴展陣列
};
```

```
// 核心加法函數
Polynomial Polynomial::Add(const Polynomial& poly) const {
    Polynomial result(capacity + poly.capacity); // 預留空間
    int i = 0, j = 0;
    while (i < terms && j < poly.terms) {
        if (termArray[i].exp == poly.termArray[j].exp) {
            result.termArray[result.terms].coef = termArray[i].coef + poly.termArray[j].coef;
            result.termArray[result.terms].exp = termArray[i].exp;
            result.terms++;
            i++;
            j++;
        }
        else if (termArray[i].exp > poly.termArray[j].exp) {
            result.termArray[result.terms++] = termArray[i++];
        }
        else {
            result.termArray[result.terms++] = poly.termArray[j++];
        }
    }
```

```
    // 合併剩餘項
    while (i < terms) result.termArray[result.terms++] = termArray[i++];
    while (j < poly.terms) result.termArray[result.terms++] = poly.termArray[j++];
    return result;
}
```

Chapter 3

效能分析

時間複雜度

加法：**O**(n+m)，其中 n 與 m 為兩個多項式的非零項數。

乘法：**O**(n*m)

評估: **O**(n)

空間複雜度

動態陣列的額外需求空間需求為 **O**(n+m)

Chapter 4

測試與驗證

1 輸入多項式 p(x)=3x2+2x+1p(x) = 3x^2 + 2x + 1p(x)=3x2+2x+1 和 q(x)=5x3+x2q(x) = 5x^3 + x^2q(x)=5x3+x2。

2 驗證加法輸出：5x3+4x2+2x+15x^3 + 4x^2 + 2x + 15x3+4x2+2x+1。

3 驗證乘法輸出：15x5+6x4+5x3+2x2+x15x^5 + 6x^4 + 5x^3 + 2x^2 + x15x5+6x4+5x3+2x2+x。

4 驗證評估：當 x=2x = 2x=2，結果為 p(2)=3(2)2+2(2)+1=17p(2) = 3(2)^2 + 2(2) + 1 = 17p(2)=3(2)2+2(2)+1=17。

效能量測

小規模測試（項數 < 10）。

大規模測試（項數 > 10,000）。

Chapter 5

心得與討論

在這次的學習中，掌握了陣列的資料結構設計，且深入理解記憶體的操作，在改進方面可以利用”vector”來簡化記憶體管理。