

資料結構HW 3

姓名:紀宇駿

August 24, 2024

CONTENTS

1. 解題說明

2 . 演算法設計與實作

3 . 效能分析

4 . 測試與驗證

5. 效能量測

6. 心得

CHAPTER 1

解題說明

設計一個Polynomial，進行加法、乘法 處理多項式的輸入 輸出使用運算符號。

CHAPTER 2

演算法設計與實作

```
1  #include <iostream>
2  #include <vector>
3  #include <cmath>
4  #include <algorithm>
5  using namespace std;
6
7  class PolyTerm {
8  public:
9      float coefficient;
10     int exponent;
11
12     PolyTerm(float coef = 0, int exp = 0) : coefficient(coef), exponent(exp) {}
13 };
14
15 class Poly {
16 private:
17     vector<PolyTerm> terms;
18
19 public:
20     Poly() {}
21
22     void addTerm(float coef, int exp) {
23         for (auto& term : terms) {
24             if (term.exponent == exp) {
25                 term.coefficient += coef;
26                 return;
27             }
28         }
29         terms.push_back(PolyTerm(coef, exp));
30     }
31
32     friend istream& operator>>(istream& input, Poly& p);
33     friend ostream& operator<<(ostream& output, const Poly& p);
34
35     Poly operator+(const Poly& other) const;
36     Poly operator-(const Poly& other) const;
37     Poly operator*(const Poly& other) const;
38
39     float evaluate(float x) const;
40 };
41
42 istream& operator>>(istream& input, Poly& p) {
43     int numTerms;
```

```

43     int numTerms;
44     cout << "Enter number of terms: ";
45     input >> numTerms;
46
47     for (int i = 0; i < numTerms; ++i) {
48         float coef;
49         int exp;
50         cout << "Enter coefficient and exponent: ";
51         input >> coef >> exp;
52         p.addTerm(coef, exp);
53     }
54
55     sort(p.terms.begin(), p.terms.end(), [](const PolyTerm& a, const PolyTerm& b) {
56         return a.exponent > b.exponent;
57     });
58
59     return input;
60 }
61
62 ostream& operator<<(ostream& output, const Poly& p) {
63     for (size_t i = 0; i < p.terms.size(); ++i) {
64         output << p.terms[i].coefficient << "x^" << p.terms[i].exponent;
65         if (i != p.terms.size() - 1) output << " + ";
66     }
67     return output;
68 }
69
70 Poly Poly::operator+(const Poly& other) const {
71     Poly result = *this;
72     for (const auto& term : other.terms) {
73         result.addTerm(term.coefficient, term.exponent);
74     }
75     return result;
76 }
77
78 Poly Poly::operator-(const Poly& other) const {
79     Poly result = *this;
80     for (const auto& term : other.terms) {
81         result.addTerm(-term.coefficient, term.exponent);
82     }
83     return result;
84 }
85
86 Poly Poly::operator/(const Poly& other) const {

```

```

85
86 Poly Poly::operator*(const Poly& other) const {
87     Poly result;
88     for (const auto& term1 : terms) {
89         for (const auto& term2 : other.terms) {
90             result.addTerm(term1.coefficient * term2.coefficient, term1.exponent + term2.exponent);
91         }
92     }
93     return result;
94 }
95
96 float Poly::evaluate(float x) const {
97     float result = 0.0;
98     for (const auto& term : terms) {
99         result += term.coefficient * pow(x, term.exponent);
100     }
101     return result;
102 }
103
104 int main() {
105     Poly poly1, poly2;
106
107     cout << "Input the first polynomial:\n";
108     cin >> poly1;
109     cout << "Input the second polynomial:\n";
110     cin >> poly2;
111
112     cout << "Polynomial 1: " << poly1 << endl;
113     cout << "Polynomial 2: " << poly2 << endl;
114
115     Poly sum = poly1 + poly2;
116     Poly diff = poly1 - poly2;
117     Poly product = poly1 * poly2;
118
119     cout << "Sum: " << sum << endl;
120     cout << "Difference: " << diff << endl;
121     cout << "Product: " << product << endl;
122
123     return 0;
124 }
125

```

CHAPTER 3

效能分析

Time Complexity

$O(n*m + m \log m)$

Space Complexity

$O(m + n)$

CHAPTER 4

測試與驗證

MICROSOFT VISUAL STUDIO 開發工具

```
Input the first polynomial:
Enter number of terms: 2
Enter coefficient and exponent: 1
2
Enter coefficient and exponent: 2
4
Input the second polynomial:
Enter number of terms: 5
Enter coefficient and exponent: 6
5
Enter coefficient and exponent: 5
4
Enter coefficient and exponent: 4
4
Enter coefficient and exponent: 4
4
Enter coefficient and exponent: 4
4
Polynomial 1: 2x^4 + 1x^2
Polynomial 2: 6x^5 + 17x^4
Sum: 19x^4 + 1x^2 + 6x^5
Difference: -15x^4 + 1x^2 + -6x^5
Product: 12x^9 + 34x^8 + 6x^7 + 17x^6
```

C:\資料結構暑假修\hw3\hw3\Debug\hw3.exe (處理序 27120) 已結束，出現代碼 0。
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [偵錯停止時，自動關閉主控台]。
按任意鍵關閉此視窗...

CHAPTER 5

效能量測

CHAPTER 6

心得

這次作業有點期末專題的感覺 算是學過的東西做個整理收穫很多

程式及演算法過程參考chat GPT