

# 多項式相乘



資訊四丙

D0683497

柯利韋

## 目錄

一、	資料結構.....	2
1.	時間複雜度.....	2
2.	多項式的一項.....	2
3.	多項式.....	3
二、	演算法說明.....	4
三、	程式原始碼.....	5
四、	執行結果.....	12
五、	心得.....	12

## 一、資料結構

### 1. 時間複雜度

- 排序： $O(n^2)$
- 輸出： $O(n)$
- 相乘： $O(n^2)$
- 總和： $O(n^2)$

### 2. 多項式的一項

```
struct Node {  
    int coef; // 係數  
    int exp;  // 次方  
};
```

### 3. 多項式

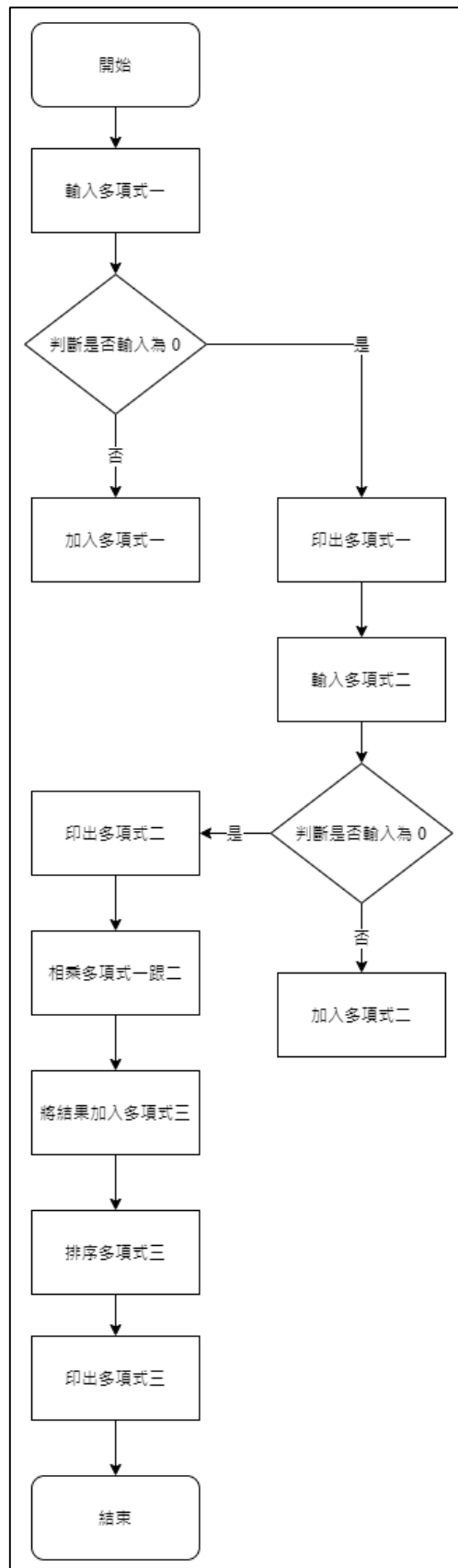
```
struct Poly
{
    // 開始位置、結束位置、下次要放的位置
    int list_1_start, list_1_end, list_1_next;
    int list_2_start, list_2_end, list_2_next;
    int list_3_start, list_3_end, list_3_next; // 放相乘結果

    Node list[MAX_SIZE];

    Poly()
    {
        list_1_start = list_1_end = list_1_next = 0;
        list_2_start = list_2_end = list_2_next = 0;
        list_3_start = list_3_end = list_3_next = 0;
    }

    void add_list_1(int, int);
    void add_list_2(int, int);
    void add_list_3(int, int);
    void print_list_1();
    void print_list_2();
    void print_list_3();
    void mul_list_1_2_to_3();
    void bubble_sort_list_3();
};
```

## 二、演算法說明



### 三、程式原始碼

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #define MAX_SIZE 10000
4
5 struct Node {
6     int coef; // 係數
7     int exp; // 次方
8 };
9
10 struct Poly
11 {
12     // 開始位置、結束位置、下次要放的位置
13     int list_1_start, list_1_end, list_1_next;
14     int list_2_start, list_2_end, list_2_next;
15     int list_3_start, list_3_end, list_3_next; // 放相乘結果
16
17     Node list[MAX_SIZE];
18
19     Poly()
20     {
21         list_1_start = list_1_end = list_1_next = 0;
22         list_2_start = list_2_end = list_2_next = 0;
23         list_3_start = list_3_end = list_3_next = 0;
24     }
25
26     void add_list_1(int, int);
27     void add_list_2(int, int);
28     void add_list_3(int, int);
29     void print_list_1();
30     void print_list_2();
31     void print_list_3();
32     void mul_list_1_2_to_3();
33     void bubble_sort_list_3();
34 };
35
36 void Poly::add_list_1(int coef, int exp)
37 {
38     if (coef == 0) // 係數為零次項
39     {
40         return;
41     }
42
43     list[list_1_next].coef = coef;
44     list[list_1_next].exp = exp;
45     list_1_end = list_1_next;
46     list_1_next++;
47     return;
48 }
49
```

```

50 void Poly::add_list_2(int coef, int exp)
51 {
52     if (coef == 0) // 係數為零次項
53     {
54         return;
55     }
56
57     if (list_2_next == 0) // 第一次放
58     {
59         list_2_start = list_1_next;
60         list_2_next = list_1_next;
61     }
62
63     list[list_2_next].coef = coef;
64     list[list_2_next].exp = exp;
65     list_2_end = list_2_next;
66     list_2_next++;
67     return;
68 }
69
70 void Poly::add_list_3(int coef, int exp)
71 {
72     int i;
73
74     if (coef == 0) // 係數為零次項
75     {
76         return;
77     }
78
79     if (list_3_next == 0) // 第一次放
80     {
81         list_3_start = list_2_next;
82         list_3_next = list_2_next;
83     }
84
85     for (i = list_3_start; i <= list_3_end; i++)
86     {
87         if (list[i].exp == exp)
88         {
89             list[i].coef = list[i].coef + coef;
90             return;
91         }
92     }
93
94     list[list_3_next].coef = coef;
95     list[list_3_next].exp = exp;
96     list_3_end = list_3_next;
97     list_3_next++;
98     return;
99 }
100

```

```

101 /* 輸出多項式1 */
102 void Poly::print_list_1()
103 {
104     int i;
105
106     printf("--多項式1為: ");
107     for (i = 0; i <= list_1_end; i++)
108     {
109         if (list[i].coef == 1)
110         {
111             if (list[i].exp == 0)
112             {
113                 printf("1+");
114             }
115             else if (list[i].exp == 1)
116             {
117                 printf("x+");
118             }
119             else
120             {
121                 printf("x^%d+", list[i].exp);
122             }
123         }
124         else if (list[i].exp == 0)
125         {
126             printf("%d+", list[i].coef);
127         }
128         else
129         {
130             printf("%dx^%d+", list[i].coef, list[i].exp);
131         }
132     }
133
134     printf("\b \b"); // 刪除最後一個符號
135     printf("\n\n");
136
137     return;
138 }
139

```



```

140 /* 輸出多項式2 */
141 void Poly::print_list_2()
142 {
143     int i;
144
145     printf("--多項式2為: ");
146     for (i = list_2_start; i <= list_2_end; i++)
147     {
148         if (list[i].coef == 1)
149         {
150             if (list[i].exp == 0)
151             {
152                 printf("1+");
153             }
154             else if (list[i].exp == 1)
155             {
156                 printf("x+");
157             }
158             else
159             {
160                 printf("x^%d+", list[i].exp);
161             }
162         }
163         else if (list[i].exp == 0)
164         {
165             printf("%d+", list[i].coef);
166         }
167         else
168         {
169             printf("%dx^%d+", list[i].coef, list[i].exp);
170         }
171     }
172
173     printf("\b \b"); // 刪除最後一個符號
174     printf("\n\n");
175
176     return;
177 }
178

```

```

179 /* 輸出結果 */
180 void Poly::print_list_3()
181 {
182     int i;
183
184     for (i = list_3_start; i <= list_3_end; i++)
185     {
186         if (list[i].coef == 1)
187         {
188             if (list[i].exp == 0)
189             {
190                 printf("1+");
191             }
192             else if (list[i].exp == 1)
193             {
194                 printf("x+");
195             }
196             else
197             {
198                 printf("x^%d+", list[i].exp);
199             }
200         }
201         else if (list[i].exp == 0)
202         {
203             printf("%d+", list[i].coef);
204         }
205         else
206         {
207             printf("%dx^%d+", list[i].coef, list[i].exp);
208         }
209     }
210
211     printf("\b \b"); // 刪除最後一個符號
212     printf("\n\n");
213
214     return;
215 }
216
217 void Poly::mul_list_1_2_to_3()
218 {
219     int i, j;
220
221     for (i = list_1_start; i <= list_1_end; i++)
222     {
223         for (j = list_2_start; j <= list_2_end; j++)
224         {
225             add_list_3(list[i].coef * list[j].coef, list[i].exp + list[j].exp);
226         }
227     }
228
229     return;
230 }
231

```

```

232 void Poly::bubble_sort_list_3()
233 {
234     int length = list_3_end - list_3_start + 1;
235     int i;
236     Node temp;
237
238     while (length > 1)
239     {
240         length--;
241         for (i = list_3_start; i < length+list_3_start; i++) //每次循環要比較的次
數
242         {
243             if (list[i].exp > list[i+1].exp) // 比大小後交換
244             {
245                 temp = list[i];
246                 list[i] = list[i+1];
247                 list[i+1] = temp;
248             }
249         }
250     }
251
252     return;
253 }
254

```

```

255 int main()
256 {
257     Poly poly;
258     int num1, num2;
259
260     /* 輸入多項式1 */
261     while (1)
262     {
263         printf("輸入多項式1(係數 次方): ");
264         scanf("%d %d", &num1, &num2);
265
266         if (num1 == 0 && num2 == 0)
267         {
268             break;
269         }
270
271         poly.add_list_1(num1, num2);
272     }
273
274     /* 輸出多項式1 */
275     poly.print_list_1();
276
277     /* 輸入多項式2 */
278     while (1)
279     {
280         printf("輸入多項式2(係數 次方): ");
281         scanf("%d %d", &num1, &num2);
282
283         if (num1 == 0 && num2 == 0)
284         {
285             break;
286         }
287
288         poly.add_list_2(num1, num2);
289     }
290
291     /* 輸出多項式2 */
292     poly.print_list_2();
293
294     /* 相乘 */
295     poly.mul_list_1_2_to_3();
296
297     /* 排序 */
298     poly.bubble_sort_list_3();
299
300     /* 輸出結果 */
301     poly.print_list_3();
302
303     return 0;
304 }

```

#### 四、執行結果

```
k139@k139: /mnt/c/Users/ff129/Desktop/FCU_Data_Structure/HW2&3/HW2$ ./main.out
輸入多項式1(係數 次方): 0 1
輸入多項式1(係數 次方): 1 2
輸入多項式1(係數 次方): 2 3
輸入多項式1(係數 次方): 3 4
輸入多項式1(係數 次方): 0 0
--多項式1為:  $x^2+2x^3+3x^4$ 

輸入多項式2(係數 次方): 4 5
輸入多項式2(係數 次方): 5 6
輸入多項式2(係數 次方): 6 7
輸入多項式2(係數 次方): 7 0
輸入多項式2(係數 次方): 0 0
--多項式2為:  $4x^5+5x^6+6x^7+7$ 

 $7x^2+14x^3+21x^4+4x^7+13x^8+28x^9+27x^{10}+18x^{11}$ 
```

#### 五、心得

因為實習課已經有做過類似得題目，基本上沒有遇到甚麼困難。