

자료구조응용

05. 다항식 더하기, 전치행렬 구하기 (10점)

2022.3.21.

- 두 다항식 정보를 파일로부터 입력받아 다항식 더하기를 수행한 후 그 결과를 출력하는 프로그램을 작성하라. 단, 다음 구조체 정의, 프로그램 2.6 및 2.7을 사용하여 구현하라.(5점)

```
#define MAX_TERMS 100
typedef struct
{
    float coef;
    int exp;
} term;

term terms[MAX_TERMS];
int avail = 0;

void padd(int startA,int finishA,int startB, int finishB,
          int *startD,int *finishD)
/* add A(x) and B(x) to obtain D(x) */
{
    float coefficient;
    *startD = avail;
    while (startA <= finishA && startB <= finishB)
        switch(COMPARE(terms[startA].expon,
                       terms[startB].expon)) {
            case -1: /* a expon < b expon */
                attach(terms[startB].coef,terms[startB].expon);
                startB++;
                break;
            case 0: /* equal exponents */
                coefficient = terms[startA].coef +
                               terms[startB].coef;
                if (coefficient)
                    attach(coefficient,terms[startA].expon);
                startA++;
                startB++;
                break;
            case 1: /* a expon > b expon */
                attach(terms[startA].coef,terms[startA].expon);
                startA++;
        }
    /* add in remaining terms of A(x) */
    for(; startA <= finishA; startA++)
        attach(terms[startA].coef,terms[startA].expon);
    /* add in remaining terms of B(x) */
    for( ; startB <= finishB; startB++)
        attach(terms[startB].coef, terms[startB].expon);
    *finishD = avail-1;
}
```

Program 2.6: Function to add two polynomials

```

void attach(float coefficient, int exponent)
/* add a new term to the polynomial */
if (avail >= MAX_TERMS) {
    fprintf(stderr,"Too many terms in the polynomial\n");
    exit(EXIT_FAILURE);
}
terms[avail].coef = coefficient;
terms[avail++].expon = exponent;
}

```

Program 2.7: Function to add a new term

$$A(x) = 2x^{1000} + 1 \quad \text{and} \quad B(x) = x^4 + 10x^3 + 3x^2 + 1$$

		terms[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
		startA	finishA	startB			finishB	avail					
<i>coeff</i>		2	1	1	10	3	1						
<i>exp</i>		1000	0	4	3	2	0						
		startA	finishA	startB			finishB	startD	avail				
<i>coeff</i>		2	1	1	10	3	1	2					
<i>exp</i>		1000	0	4	3	2	0	1000					
		startA	finishA	startB			finishB	startD	avail				
<i>coeff</i>		2	1	1	10	3	1	2	1	10			
<i>exp</i>		1000	0	4	3	2	0	1000	4	3			
		startA	finishA	startB			finishB	startD	avail				
<i>coeff</i>		2	1	1	10	3	1	2	1	10			
<i>exp</i>		1000	0	4	3	2	0	1000	4	3			
		startA	finishA	startB			finishB	startD	avail				
<i>coeff</i>		2	1	1	10	3	1	2	1	10	3	2	
<i>exp</i>		1000	0	4	3	2	0	1000	4	3	2	0	

< 두 다항식의 덧셈 과정 >

[입력파일(input.txt)]

$$A(x) = 2x^{1000} + 1 \quad \text{and} \quad B(x) = x^4 + 10x^3 + 3x^2 + 1$$

2 4	2 1000
1 0	
1 4	
10 3	
3 2	
1 0	

* 첫 줄의 2 4는 각 다항식의 항 개수임

[함수원형]

① void inputPoly(char *fname, int *startA, int *finishA, int *startB, int *finishB);
- 파일(input.txt)로부터 데이터를 입력받아 2개의 다항식을 구성

② void printPoly(int start, int finish);
- 하나의 다항식을 전달받아 화면으로 출력

[실행예]

```
C:\> C:\WINDOWS\system32\cmd.exe - □ X  
two input polynomials ...  
 $A(x) = 2.0x^{1000} + 1.0$   
 $B(x) = 1.0x^4 + 10.0x^3 + 3.0x^2 + 1.0$   
the result of polynomial addition ...  
 $D(x) = A(x) + B(x) =$   
 $2.0x^{1000} + 1.0x^4 + 10.0x^3 + 3.0x^2 + 2.0$   
계속하려면 아무 키나 누르십시오 . . .
```

2. 다음과 같은 조건으로 희소행렬의 전치행렬을 구하는 프로그램을 작성하라. (5점)

```

void transpose(term a[], term b[])
{ /* b is set to the transpose of a */
    int n,i,j, currentb;
    n = a[0].value;           /* total number of elements */
    b[0].row = a[0].col; /* rows in b = columns in a */
    b[0].col = a[0].row; /* columns in b = rows in a */
    b[0].value = n;
    if (n > 0) { /* non zero matrix */
        currentb = 1;
        for (i = 0; i < a[0].col; i++)
            /* transpose by the columns in a */
            for (j = 1; j <= n; j++)
                /* find elements from the current column */
                if (a[j].col == i) {
                    /* element is in current column, add it to b */
                    b[currentb].row = a[j].col;
                    b[currentb].col = a[j].row;
                    b[currentb].value = a[j].value;
                    currentb++;
                }
        }
    }
}

```

Program 2.8: Transpose of a sparse matrix

[입출력 형식 및 실행예]

입력파일(a.txt)	출력파일(b.txt)
7 6 8	6 7 8
0 0 13	0 0 13
0 3 22	0 4 5
1 2 3	1 3 1
3 1 1	1 6 12
3 4 17	2 1 3
4 0 5	3 0 22
4 3 19	3 4 19
6 1 12	4 3 17

A	13	0	0	22	0	0
	0	0	3	0	0	0
	0	0	0	0	0	0
	0	1	0	0	17	0
	5	0	0	19	0	0
	0	0	0	0	0	0
	0	12	0	0	0	0
B	13	0	0	0	5	0
	0	0	0	1	0	12
	0	3	0	0	0	0
	22	0	0	0	19	0
	0	0	0	17	0	0
	0	0	0	0	0	0

■ 제출 형식

- 솔루션 이름 : DS 05

- 프로젝트 이름 : 1, 2

- 각 소스파일에 주석처리

“학번 이름”

“본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.”

- 실행화면을 캡쳐한 보고서를 작성 후 pdf 파일로 변환하여 솔루션 폴더에 포함

- 솔루션 정리 메뉴를 수행 후 전체 솔루션을 “학번.zip”으로 압축하여 제출

■ 주의

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!

- 1차 마감 (LMS 과제 마감일) : 수업일 자정

- 2차 마감 (LMS 과제 열람기한) : 수업 익일 자정(만점의 50%, 반올림)

- 1, 2차 마감 이외의 제출은 허용하지 않습니다. (이메일 제출 불가!)