

자료구조응용

06. 다항식더하기, 전치행렬구하기 (10점)

2020.4.1. (수)

1. 두 다항식을 입력받아 더한 후 그 결과를 출력하는 프로그램을 작성하라. (5점)

[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

[프로그램설명]

- ① 각 다항식 데이터를 파일 입력함
- ② 교재의 `padd`, `attach` 함수는 수정 없이 그대로 사용하기
- ③ 다항식을 출력하는 함수를 정의하여 사용하기

```
#define MAX_TERMS 100 /*size of terms array*/
typedef struct {
    float coef;
    int expon;
} 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

[실행 예]

```

C:\WINDOWS\system32\cmd.exe
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. 다음 희소행렬(sparse matrix)에 대한 전치행렬(transpose matrix)을 구하는 함수로 Program 2.9 fastTranspose를 구현하여 실행결과를 보여라. (5점)

[프로그램 설명]

① 파일 입력 및 출력(Figure 2.5의 예)

※ 행과 열의 크기는 최대 10까지 입력될 수 있음

입력파일(a.txt)			출력파일(b.txt)		
6	6	8	6	6	8
0	0	15	0	0	15
0	3	22	0	4	91
0	5	-15	1	1	11
1	1	11	2	1	3
1	2	3	2	5	28
2	3	-6	3	0	22
4	0	91	3	2	-6
5	2	28	5	0	-15

	row	col	value		row	col	value
a[0]	6	6	8	b[0]	6	6	8
[1]	0	0	15	[1]	0	0	15
[2]	0	3	22	[2]	0	4	91
[3]	0	5	-15	[3]	1	1	11
[4]	1	1	11	[4]	2	1	3
[5]	1	2	3	[5]	2	5	28
[6]	2	3	-6	[6]	3	0	22
[7]	4	0	91	[7]	3	2	-6
[8]	5	2	28	[8]	5	0	-15
	(a)				(b)		

Figure 2.5: Sparse matrix and its transpose stored as triples

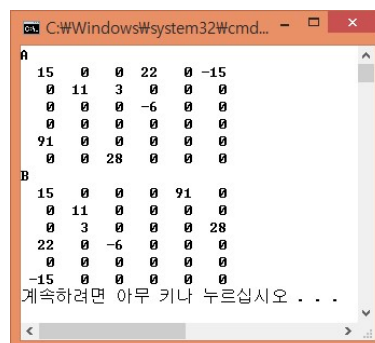
② fastTranspose 함수는 수정 없이 그대로 유지

③ a 혹은 b 행렬 하나를 전달받아 실행 예와 같이 화면출력하는 함수 정의

```
void fastTranspose(term a[], term b[])
/* the transpose of a is placed in b */
int rowTerms[MAX-COL], startingPos[MAX-COL];
int i, j, numCols = a[0].col, numTerms = a[0].value;
b[0].row = numCols; b[0].col = a[0].row;
b[0].value = numTerms;
if (numTerms > 0) { /* nonzero matrix */
    for (i = 0; i < numCols; i++)
        rowTerms[i] = 0;
    for (i = 1; i <= numTerms; i++)
        rowTerms[a[i].col]++;
    startingPos[0] = 1;
    for (i = 1; i < numCols; i++)
        startingPos[i] =
            startingPos[i-1] + rowTerms[i-1];
    for (i = 1; i <= numTerms; i++) {
        j = startingPos[a[i].col]++;
        b[j].row = a[i].col; b[j].col = a[i].row;
        b[j].value = a[i].value;
    }
}
```

Program 2.9: Fast transpose of a sparse matrix

[실행 예]



■ 제출 형식

- 솔루션 이름 : DS 06
- 프로젝트 이름 : 1, 2
- 각 소스파일에 주석처리
“학번 이름”
“본인은 이 소스파일을 복사 없이 직접 작성하였습니다.”
- 솔루션 정리 후 솔루션 폴더를 “학번.zip”으로 압축하여 과제 게시판에 제출

■ 주의

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!
- 마감 : 다음 수업일 전 자정