

자료구조 숙제 #1

2024. 1 학기

※ 제출 관련 사항

(1) 제출일 : 2024 년 10 월 11 일

(2) 제출 방법

제출 파일 명 : ds_hw_1_학번.zip (ex: ds_hw_1_20245678.zip)

(제출 파일 내 동작하는 problem_1.c, problem_2.c 포함) → C로 풀기

기한 내 cyber 캠퍼스내 제출

※ 문제 1: 두 행렬의 덧셈 → problem_1.c

두 개의 희소 행렬을 파일 입력으로 받아 (1) 두 행렬의 덧셈 결과 행렬과 (2) 두 행렬의 뺄셈 결과 행렬을 만들고 / 희소 행렬 형태로 출력하는 프로그램을 작성하시오.

이때, 덧셈과 뺄셈 연산의 코드 앞 뒤로, 계산에 걸린 시간, time measure 를 측정할 수 있는 코드를 추가로 작성 하고, 각 연산을 1000 번 반복하여 평균 연산 시간을 출력하시오.

※ 입력

matrix1.txt 와 matrix2.txt 에 각각의 행렬에 대한 정보가 주어진다. 첫째 줄에 정수 n, m 이 공백으로 구분되어 주어지고, 둘째 줄부터 n+1 번째 줄에 m 개의 원소가 공백으로 구분되어 주어진다.

※ 출력

두 행렬의 덧셈 및 뺄셈 결과를 희소 행렬 형태로 표준 출력한다. (화면에 출력한다.) 출력 첫 줄에는 행렬의 크기와 0 이 아닌 행렬의 값의 개수를 출력한다. 출력 마지막 줄에는 평균 연산 시간을 출력한다.

※ 입력 예제

9 10									
61	-45	0	0	28	0	0	74	0	3
0	0	3	15	8	0	0	45	1	0
0	1	0	9	1	1	0	1	0	0
5	9	7	0	0	0	0	100	0	2
0	0	0	1	13	0	14	15	0	0
0	78	37	1	0	1	0	0	0	0
0	0	5	52	0	0	0	59	0	5
0	0	0	0	2	3	4	0	0	0
0	0	0	0	11	1	0	0	0	1
matrix1.txt									
9 10									

□ 코드
시작시간
덧셈연산
끝시간
시작시간
뺄셈연산
끝시간

62	-42	0	0	0	0	3	0	0	2
2	0	0	14	7	0	0	44	2	0
0	0	0	5	9	7	0	0	0	0
4	19	6	0	0	20	0	99	1	1
0	0	0	0	12	13	0	0	0	0
-10	0	3	0	1	0	48	0	0	0
0	0	-5	5	11	0	1	55	0	4
0	0	0	0	20	0	2	0	0	0
0	0	0	0	1	3	0	1	0	0

matrix2.txt

※ 출력 예제

matrix1 + matrix2
 9 10 48 → 값의 개수 출력 ⇒ 0인 값은 제외.
 0 0 123
 0 1 -87
 0 4 28
 0 6 3
 0 7 74
 0 9 5
 1 0 2
 1 2 3
 1 3 29
 1 4 15
 1 7 89
 1 8 3
 2 1 1
 2 3 14
 2 4 10
 2 5 8
 2 7 1
 3 0 9
 3 1 28
 3 2 13
 3 5 20
 3 7 199
 3 8 1
 3 9 3
 4 3 1
 4 4 25
 4 5 13
 4 6 14
 4 7 15
 5 0 -10
 5 1 78
 5 2 40
 5 3 1

↓ 출력기준 → Row → C/umn

```
5 4 1
5 5 1
5 6 48
6 3 57
6 4 11
6 6 1
6 7 114
6 9 9
7 4 22
7 5 3
7 6 6
8 4 12
8 5 4
8 7 1
8 9 1
```

평균 연산 시간(1000 회 반복): 1.42 초

matrix1 - matrix2

```
9 10 49
0 0 -1
0 1 -3
0 4 28
0 6 -3
0 7 74
0 9 1
1 0 -2
1 2 3
1 3 1
1 4 1
1 7 1
1 8 -1
2 1 1
2 3 4
2 4 -8
2 5 -6
2 7 1
3 0 1
3 1 -10
3 2 1
3 5 -20
3 7 1
3 8 -1
3 9 1
4 3 1
4 4 1
```

4 5 -13
4 6 14
4 7 15
5 0 10
5 1 78
5 2 34
5 3 1
5 4 -1
5 5 1
5 6 -48
6 2 10
6 3 47
6 4 -11
6 6 -1
6 7 4
6 9 1
7 4 -18
7 5 3
7 6 2
8 4 10
8 5 -2
8 7 -1
8 9 1

평균 연산 시간(1000 회 반복): 1.42 초

※ 자료구조 및 알고리즘

```
#include <stdio.h>
#include <stdlib.h>
typedef struct {
    int row, column, value;
} Term;
#define MAX_TERMS 1024
void readMatrix(FILE* fp, Term a[]);
void printMatrix(Term a[]);
void matrixAdd(Term a[], Term b[], Term c[]);
void matrixSubtract(Term a[], Term b[], Term c[]);
int main() {
    Term a[MAX_TERMS], b[MAX_TERMS], c[MAX_TERMS];

    FILE *fp = fopen("A.txt", "r");
    readMatrix(fp, a);
    fclose(fp);

    fp = fopen("B.txt", "r");
    readMatrix(fp, b);
    fclose(fp);

    matrixAdd(a, b, c);
    printMatrix(c);

    matrixSubtract(a, b, c);
```

```
    printMatrix(c);  
    return 0;  
}
```

※ 문제 2 : 두 행렬의 곱셈

두 개의 희소 행렬을 파일 입력으로 받아 두 행렬의 곱셈 결과 행렬을 만들고 희소 행렬 형태로 출력하는 프로그램을 작성하시오. (첫 번째 행렬에 두 번째 행렬을 전치하여 곱하시오.)

이때, 책에 나온 것 처럼 fastTranspose 를 활용하며, fastTranspose 는 수업시간에 언급했듯이 startingPos 배열을 사용하지 않고 rowTerms 만을 사용해서 구현하시오.

또한, 곱셈 연산의 코드 앞 뒤로, 계산에 걸린 시간, time measure 를 측정할 수 있는 코드를 추가로 작성 하고, 각 연산을 1000 번 반복하여 평균 계산 시간을 출력하시오.

※ 입력

matrix3.txt 와 matrix4.txt 에 각각의 행렬에 대한 정보가 주어진다. 첫째 줄에 정수 n, m 이 공백으로 구분되어 주어지고, 둘째 줄부터 n+1 번째 줄에 m 개의 원소가 공백으로 구분되어 주어진다.

※ 출력

두 행렬의 곱셈 결과를 희소 행렬 형태로 표준 출력한다. (화면에 출력한다.)

출력 첫 줄에는 행렬의 크기와 0 이 아닌 행렬의 값의 개수를 출력한다. 출력 마지막 줄에는 평균 연산 시간을 출력한다.

※ 입력 예제

5 7
0 0 3 0 0 0 0
1 0 0 0 4 0 0
0 0 0 2 0 0 1
0 0 0 0 0 5 0
0 5 0 0 0 0 0

matrix3.txt

7 6
0 0 0 4 0 0
0 0 2 0 0 0
1 0 0 0 0 0
0 5 0 0 0 0
0 0 0 0 6 0
0 0 1 0 0 3
0 2 0 0 0 0

matrix4.txt

※ 출력 예제

matrix3 * matrix4

5 6 7

0 0 3

1 3 4

1 4 24

2 1 12

3 2 5

3 5 15

4 2 10

총 7개 행 → row

평균 연산 시간(1000 회 반복): 2.58 초