

2024-1학기 자료구조 1분반 과제 #3: Sparse Matrices

2024. 5. 10, 오후 5시까지

입력이 다음과 같이 주어질 때, 수업 시간에 다룬 알고리즘을 참고해 **matrix addition function (madd)**을 비롯한 **function**을 구현하시오.

(1) 입력 : 파일에 저장된 두 개의 m by n matrix

파일 A.txt에 matrix A 저장

파일 B.txt에 matrix B 저장

입력의 예

6 7	6 7
0 4 0 7 0 0 9	15 0 0 0 9 1 0 3
2 0 0 0 6 5 0	0 11 0 0 0 0 7
0 0 3 8 0 4 7	0 3 0 0 0 28 4
0 0 0 0 0 1 0	22 0 -6 0 0 0 1
1 2 3 0 0 0 8	0 0 0 0 0 0 -4
0 5 0 4 3 0 2	3 -5 1 0 0 -2 0
A.txt	B.txt

각 파일에 저장된 matrix

- 첫째 줄 : matrix의 행의 수와 열의 수를 나타내는 두 개의 양의 정수 m과 n
- 둘째 줄 ~ (m+1)째 줄 : i번째 줄에 matrix의 (i-1)번째 행의 원소들을 나타내는 n개의 정수들

(2) 자료구조와 알고리즘 (4.7절의 선언과 함수들 참고)

```
#define MAX_SIZE 50
typedef enum {head, entry} tagfield;
typedef struct matrix_node *matrix_pointer;
typedef struct entry_node {
    int row;
    int col;
    int value;
};
typedef struct matrix_node {
    matrix_pointer down;
    matrix_pointer right;
    tagfield tag;
};
```

```

        union {
            matrix_pointer next;
            entry_node entry;
        } u;
matrix_pointer hdnnode[MAX_SIZE];

matrix_pointer mread(FILE*); // file에서 matrix 자료를 읽어 리스트에 저장한다.
void mwrite(matrix_pointer); // 행렬을 적절한 포맷 (2차원 행렬형태)으로 출력한다.
void merase(matrix_pointer*); // 행렬의 모든 노드를 지운다.
matrix_pointer madd(matrix_pointer, matrix_pointer); // matrix addition

void main()
{
    matrix_pointer a, b, d;

    파일 오픈한다.
    a = mread(FILE*);
    b = mread(FILE*);
    mwrite(a);
    mwrite(b);
    d = madd(a, b);
    mwrite(d);
    merase(&d);
    merase(&a);
    merase(&b);
    mwrite(a);
    mwrite(b);
    mwrite(d);
}

```

※ 제출 관련 사항

(1) 제출일 : 2024. 5. 10, 오후 5시까지

(2) 제출 방법

-하드 카피

코드(학번_madd.c)와 본인의 문제 해결 방법을 포함한 레포트를 제출하시오.

-소프트 카피

sogang2024.1.1.ds@gmail.com로 학번_madd.c를 보내시오.