2017-2학기 자료구조 홍의경 교수님 / 2016920060 컴퓨터과학부 최형진

```c
/* Sparse Matrix Sum
 * 2017-2 Data Structure
 * Written by Choe Hyeong Jin, Dept. of Computer Science, Univ. of Seoul
 */

#include <stdio.h>
#define ROWS 6
#define COLS 6
#define MAX_TERMS 36

typedef struct{
    int row; // 행 위치
    int col; // 열 위치
    int value; // 요소 값
} element; // 0이 아닌 요소 구조체.

typedef struct SparseMatrix{
    element data[MAX_TERMS]; // 0이 아닌 요소들 저장하는 배열
    int rows; // 행 개수
    int cols; // 열 개수
    int terms; // 요소 개수
} SparseMatrix; // 희소 행렬 표현 방법 #2(0이 아닌 요소들만 저장) 을 이용한 구조체.

void print_sparse_matrix(SparseMatrix x); // 희소 행렬 출력 함수
SparseMatrix sparse_matrix_add2(SparseMatrix a, SparseMatrix b); // 희소 행렬 두 개의 합을 리턴하는 함수

int main(void){
    SparseMatrix a = { {{0, 0, 1}, {1, 1, 1}, {2, 1, 1}, {3, 2, 1}, {4, 3, 1}, {5, 1, 2}, {5, 3, -1}}, 6, 6, 7 };
    SparseMatrix b = { {{0, 0, 1}, {1, 0, 1}, {2, 1, 1}, {3, 2, 1}, {4, 0, 1}, {5, 0, 1}}, 6, 6, 6 };
    SparseMatrix c = sparse_matrix_add2(a, b); // 희소 행렬 a, b을 합한 결과를 저장.
    // 덧셈 과정 출력. a + b = c
    print_sparse_matrix(a);
    printf("\t+\n");
    print_sparse_matrix(b);
    printf("\t=\n");
    print_sparse_matrix(c);
    return 0;
}

void print_sparse_matrix(SparseMatrix x){ // 희소 행렬 출력 함수
    int row, col, idx = 0;
    // 행 우선 출력
    for(row = 0;row < x.rows;row++){ // 행
        printf("|");
        for(col = 0;col < x.cols;col++){ // 열
            // 명시된 요소의 개수를 넘기지 않으면서 0이 아닌 요소가 있다면
            if( !(idx >= x.terms) && (x.data[idx].row == row) && (x.data[idx].col == col) ){
                printf("%2d ", x.data[idx].value); // 출력
                idx++; // 요소 인덱스 증가
            }
            else printf("%2d ", 0); // 0이 아닌 요소가 없다면 0 출력
        }
        printf("|\n"); // 행마다 개행
    }
    return; // 함수 종료
}

SparseMatrix sparse_matrix_add2(SparseMatrix a, SparseMatrix b){
    SparseMatrix c;
    int ca=0, cb=0, cc=0; // 각 희소 행렬의 요소를 가리키는 인덱스
    // 희소 행렬 a와 희소 행렬 b의 크기가 서로 같은지 확인
    if( a.rows != b.rows || a.cols != b.cols ){
        fprintf(stderr, "Sparse matrix size error!\n");
        exit(1);
```

```
65.     }
66.     // 행렬 정보 초기화.
67.     c.rows = a.rows; // a, b 와 같은 행 수
68.     c.cols = a.cols; // a, b 와 같은 열 수
69.     c.terms = 0; // 아직 아무 값도 처리하지 않았으므로 0 으로 초기화.
70.     while( ca < a.terms && cb < b.terms ){
71.         //각 항목의 순차적인 번호를 계산한다.
72.         int idx_a = a.data[ca].row * a.cols + a.data[ca].col;
73.         int idx_b = b.data[cb].row * b.cols + b.data[cb].col;
74.         if( idx_a < idx_b ){
75.             // 희소 행렬 a 의 항목이 앞에 있으면
76.             c.data[cc++] = a.data[ca++]; // a 의 값을 c 에 저장 후 a 와 c 의 인덱스 증가
77.         }
78.         else if( idx_a == idx_b ){
79.             // 희소 행렬 a 의 항목과 희소 행렬 b 의 항목이 같은 위치라면
80.             if( (a.data[ca].value + b.data[cb].value)!=0 ){ // 두 항목의 합이 0 이 아니라면
81.                 c.data[cc].row = a.data[ca].row; // 같은 행 위치에
82.                 c.data[cc].col = a.data[ca].col; // 같은 열 위치에
83.                 c.data[cc++].value = a.data[ca++].value + b.data[cb++].value; // a 와 b 의 합을 저
    장하고 a, b, c 인덱스 증가
84.             }
85.             else{ // 두 항목의 합이 0 이라면
86.                 ca++; cb++; // a 와 b 의 인덱스만 증가
87.             }
88.         }
89.         else // b 배열 항목이 앞에 있음.
90.             c.data[cc++] = b.data[cb++];  // b 의 값을 c 에 저장 후 b 와 c 의 인덱스 증가
91.     }
92.     // 배열 a 와 b 에 남아있는 항들을 배열 c 로 옮긴다.
93.     while( ca < a.terms ) c.data[cc++] = a.data[ca++];
94.     while( cb < b.terms ) c.data[cc++] = b.data[cb++];
95.     c.terms = cc; // cc 의 가장 최근 값이 c 의 항목 개수다.
96.     return c;
97. }
```

```
C:₩WINDOWS₩system32₩cmd.exe
¦ 1   0   0   0   0   0 ¦
¦ 0   1   0   0   0   0 ¦
¦ 0   1   0   0   0   0 ¦
¦ 0   0   1   0   0   0 ¦
¦ 0   0   0   1   0   0 ¦
¦ 0   2   0  -1   0   0 ¦
          +
¦ 1   0   0   0   0   0 ¦
¦ 1   0   0   0   0   0 ¦
¦ 0   1   0   0   0   0 ¦
¦ 0   0   1   0   0   0 ¦
¦ 1   0   0   0   0   0 ¦
¦ 1   0   0   0   0   0 ¦
          =
¦ 2   0   0   0   0   0 ¦
¦ 1   1   0   0   0   0 ¦
¦ 0   2   0   0   0   0 ¦
¦ 0   0   2   0   0   0 ¦
¦ 1   0   0   1   0   0 ¦
¦ 1   2   0  -1   0   0 ¦

C:\Users\lenovo\Google 드라이브\대학교\2-2\자료구조_홍의경교수님>
```