

# 자료구조응용

## 06. 전치행렬, 스택 (10점)

2022.3.23.

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

[프로그램 설명]

### (1) 희소행렬의 표현

```
#define MAX_TERMS 101

// 희소행렬의 0아닌 항
typedef struct
{
    int row;
    int col;
    int value;
} term;

// 희소행렬 a, b
term a[MAX_TERMS]; // a[0] : 행 크기, 열 크기, 0아닌 항의 개수
term b[MAX_TERMS]; // a의 전치행렬 b
```

### (2) 전치행렬 구하기 (프로그램 2.9)

- ① rowTerms 배열 초기화 : a에 대해 열 인덱스 별로 빈도 계산
- ② startingPos 배열 초기화 : b에 대해 새로운 행의 시작위치 계산
- ③  $a(i, j) \rightarrow b(j, i)$  수행

$\begin{bmatrix} 13 & 0 & 0 & 22 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 17 & 0 \\ 5 & 0 & 0 & 19 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 13 & 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 12 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 22 & 0 & 0 & 0 & 19 & 0 & 0 \\ 0 & 0 & 0 & 17 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
$a[0] \quad 7 \quad 6 \quad 8$ [1] 0 0   13 [2] 0 3 22 [3] 1 2 3      ① <i>rowTerms</i> [0] [1] [2] [3] [4] [5] [4] 3 1 1      ② <i>startPosition</i> 2   2 1 2 1 0 [5] 3 4 17 $\xrightarrow{\text{fastTranspose}}$ [1] 0 0 13 [6] 4 0 5 [7] 4 3 19 [8] 6 1 12	$b[0] \quad 6 \quad 7 \quad 8$ [1] 0 0 13 [2] 0 4 5 [3] 1 3 1 [4] 1 6 12 [5] 2 1 3 [6] 3 0 22 [7] 3 4 19 [8] 4 3 17

< 희소행렬의 표현과 전치행렬 구하기 >

---

```

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;
        ① { for (i = 1; i <= numTerms; i++)
            rowTerms[a[i].col]++;
            startingPos[0] = 1;
            ② { for (i = 1; i < numCols; i++)
                startingPos[i] =
                    startingPos[i-1] + rowTerms[i-1];
                for (j = 1; j <= numTerms; j++) {
                    ③ { 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

### [함수원형]

```
void printMatrix( term matrix[] );
```

- 희소행렬 하나를 전달받아 실행 예와 같이 화면으로 출력하는 함수

### [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

2. [정적할당배열의 스택] 다음과 같은 스택을 생성하고 실행 예와 같이 수행되는 프로그램을 작성하라. 이를 위해, push, pop, stackEmpty, stackFull 함수를 구현하여야 한다.(5점)

[자료형과 함수의 정의]

```
#define MAX_STACK_SIZE 5
typedef struct {
    int id;                                // unique id
    char name[MAX_NAME_SIZE];      // last name
} element;
element stack[MAX_STACK_SIZE];
int top = -1;
```

---

```
void push(element item)
{ /* add an item to the global stack */
    if (top >= MAX_STACK_SIZE-1)
        stackFull();
    stack[++top] = item;
}
```

---

**Program 3.1:** Add an item to a stack

---

```
element pop()
{ /* delete and return the top element from the stack */
    if (top == -1)
        return stackEmpty(); /* returns an error key */
    return stack[top--];
}
```

---

**Program 3.2:** Delete from a stack

---

```
void stackFull()
{
    fprintf(stderr, "Stack is full, cannot add element");
    exit(EXIT_FAILURE);
}
```

---

**Program 3.3:** Stack full

[구현조건]

- ① 사용자입력으로부터 데이터 추출을 위해 gets, strtok, strcmp, sscanf, strlen 등을 사용
- ② push, pop 함수는 교재 코드를 수정 없이 그대로 사용할 것
- ③ stackFull은 코드를 수정하되, pop()을 사용하여 현재 스택요소들을 출력해 준 후 exit(EXIT\_FAILURE)을 호출하도록 구현할 것  
※ EXIT\_FAILURE를 사용하기 위해 stdlib.h를 인클루드해야 함
- ④ stackEmpty는 에러메시지를 출력한 후 임의의 에러키(ex: -1)를 가지는 element형 변수를 반환
- ⑤ 잘못된 커맨드를 입력하면 에러 메시지를 출력 후 다시 사용자 입력을 받도록 함

```

int main(void)
{
    char input[80];           // input a line of string
    char *delimiter = " \n";   // delimiter for tokenizing
    char *op = NULL;          // push or pop
    element student;
    int cnt;
    ...
    while( 1 )
    {
        gets(input);
        op = strtok(input, delimiter);

        if( !strcmp(op, "push"))
        {
            sscanf(input + strlen(op)+1, "%d%s", &student.id, student.name);
            push(student);
        }
        else if( !strcmp(op, "pop"))
        {
            element item;
            item = pop();
            if ( item.id == -1 )
                exit(EXIT_FAILURE);
        }
        else if( !strcmp(op, "quit"))
            break;
        else
            printf("wrong command! try again!\n");
    }// while

    return 0;
}

```

### [ 실행 예 ]

The screenshots show the execution of a C program for stack operations. The program uses a stack of size 5 to store student information (id and name). The user can perform push, pop, add, and quit operations.

- Screenshot 1:** The stack starts empty. Elements 1 through 8 are pushed onto the stack. The stack is then full, and no more elements can be added.
- Screenshot 2:** The stack is empty. A delete operation is attempted, which fails with an error message. Then, a pop operation is performed, which also fails because the stack is empty. Finally, a quit operation is performed.
- Screenshot 3:** The stack is empty. A quit operation is performed, exiting the program.

## ■ 제출 형식

- 솔루션 이름 : DS 06
- 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! )