

자료 구조 Lab 004 :

Lab20004.zip : LabTest.java, lab004.java, lab.in, lab.out, lab004.pdf

제출

lab004.java 를 학번.java 로 변경하여 이 파일 한 개만 제출할 것.

이번 Lab은 희소 행렬 (Sparse Matrix) 자료구조가 정의된 위 .java 파일을 분석하고, 아직 구현이 덜 된 부분을 구현하는 것이다.

수행 예는 다음과 같다.

```
선택 sanghwan@PC: ~/dbox/classes201/ds/lab20/lab20004
sanghwan@PC:~/dbox/classes201/ds/lab20/lab20004$ java LabTest
Init SparseMatrix a
Enter the number of rows and columns :
3 3
Enter a term (row, col, value)
1 1 1
Enter a term (row, col, value)
2 3 2
Enter a term (row, col, value)
3 2 3
Enter a term (row, col, value)
-1
Content of smArray :
[0] 1 1 1
[1] 2 3 2
[2] 3 2 3
=====

Init SparseMatrix b
Enter the number of rows and columns :
3 3
Enter a term (row, col, value)
1 1 2
Enter a term (row, col, value)
3 2 1
Enter a term (row, col, value)
-1
Content of smArray :
[0] 1 1 2
[1] 3 2 1
=====

c = a + b
Content of smArray :
[0] 1 1 3
[1] 2 3 2
[2] 3 2 4
=====

Continue (y/n)
```

사용자로부터 2 개의 행렬을 0이 아닌 term만을 입력 받아서 초기화를 시킨다. 입력 순서는 row major로서 행 번호와 열번호가 작은 순서로 입력한다. 새로운 term을 입력할 때 -1을 입력하면 그 행렬에 대한 입력이 끝난다.

두 개의 행렬이 입력되면 두 행렬을 더해서 새로운 행렬을 만들고, 그 행렬의 smArray 상태를 보여준다.

위 예는 다음 행렬의 덧셈을 수행한 예이다.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 3 & 0 \end{pmatrix} + \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 4 & 0 \end{pmatrix}$$

이를 위해 구현이 필요한 부분은 아래의 1개의 함수이다.

- `SparseMatrix Add(SparseMatrix b);`

Polynomial ADD를 참조하면 되는데, lab004.java 내에 있는 `int Compare(MatrixTerm a, MatrixTerm b)`를 활용하여 **두 개의 원소 중 누가 앞에 나와야 하는지를 알아내면** 쉽게 구현할 수 있다.

참고로 구현을 할 때는 LabTest.java의 main 함수를 살펴보면서 lab004.java를 수정하는 것이 좋다.

프로그램 테스트

컴파일

```
$ javac lab004.java LabTest.java
```

실행

```
$ java LabTest
```

주어진 **input**으로 실행

```
$ java LabTest < lab.in
```

주어진 **output**과 비교

```
$ java LabTest < lab.in > abc
$ diff abc lab.out
```