

자료 구조 HW #2 (30점)

제출일 : 2019년 5월 12일 일요일 (eCampus)

제출물 : 학번.java 파일 하나

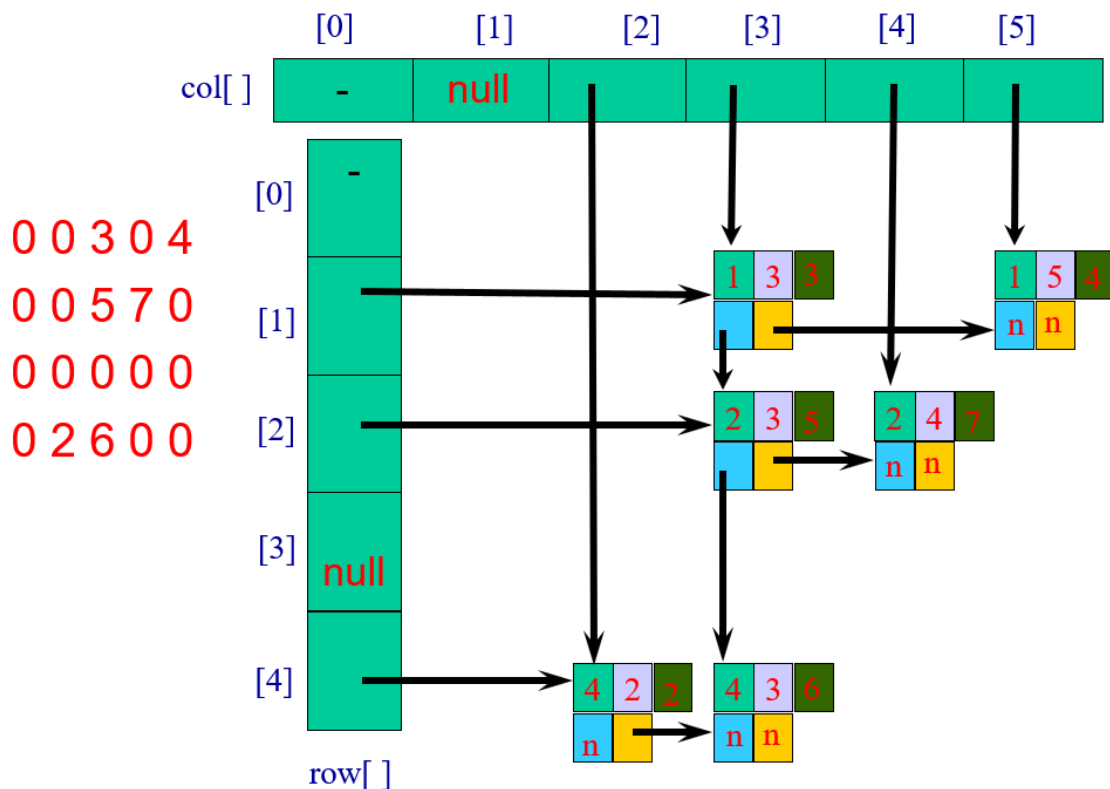
Hard copy 제출 필요 없음

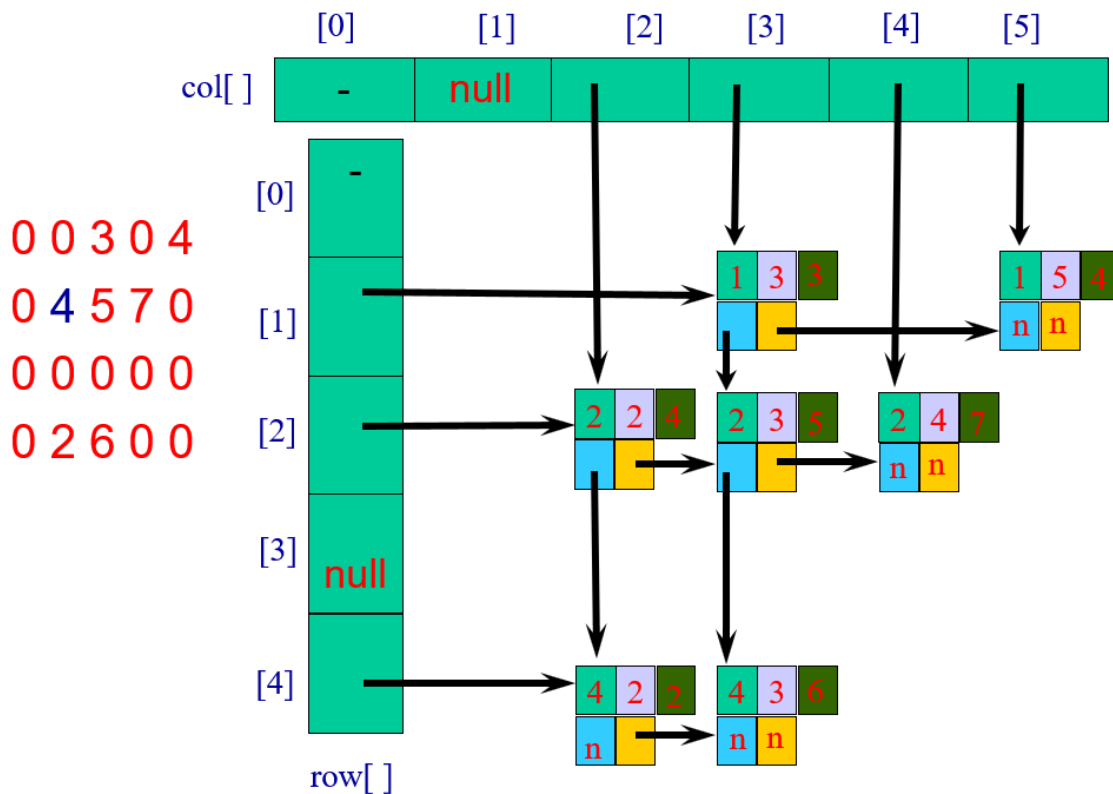
hw2.zip 파일 : LabTest.java **LinkedMatrix.java** hw2.java lab.in lab.out hw2.pdf

제출

hw2.java 를 학번.java 로 변경하여 이 파일 한 개만 제출할 것.

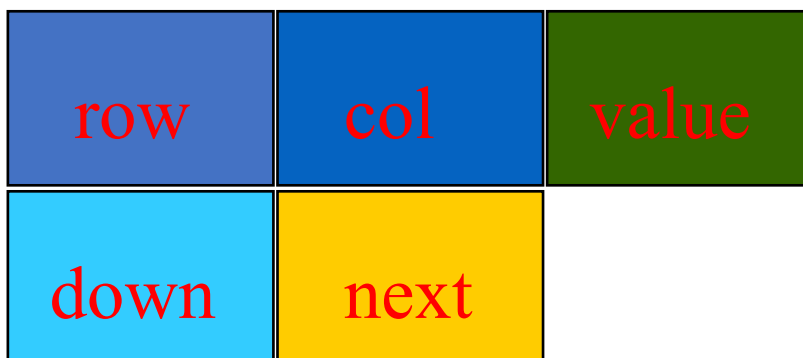
다음은 Linked List를 이용하여 Sparse Matrix를 구현하는 내용이다. 데이터 저장 구조는 다음과 같다. 0이 아닌 원소들만 체인 형태로 유지하는 것이다.





hw2.java 내에 정의된 SparseMatrix 라는 클래스는 LinkedMatrix 라는 클래스를 부모 클래스로 하여 구현된다. LinkedMatrix는 `row[]`와 `col[]` 이라는 배열을 가지고 있고, 이 배열의 원소는 위의 그림에서와 같이 각 행과 열의 첫 번째 노드의 참조를 가지고 있다. 즉 행 력의 행 `i`에 대한 체인은 `row[i]`로부터 시작하고, 열 `j`에 대한 체인은 `col[j]`로부터 시작한다. 따라서 각 배열의 0번 원소는 사용하지 않는다.

각 노드는 다음과 같은 구조를 가지고 있다. 또한 이 내용은 LinkedMatrix.java에 MatrixNode라는 클래스로 구현되어 있다. 그리고 LinkedMatrix 클래스에는 행렬을 행우선 및 열우선으로 출력하는 함수가 구현되어 있다.



```

class MatrixNode {
    int row;    // row
    int col;    // column
    int value;  // value
    MatrixNode down;    // down reference
    MatrixNode next;    // next reference

    MatrixNode(final int r, final int c, final int v) {
        row = r;
        col = c;
        value = v;
    };
};

```

LinkedMatrix 클래스를 자세히 분석하면 많은 도움이 될 것이다.

이러한 구조로 된 Sparse Matrix를 사용자로 부터 입력을 받아서 Matrix 덧셈 및 간단한 원소 값 변경을 수행하는 작업을 한다. 아래는 그 수행 예이다.

우선 행렬 a를 입력받는데, 먼저 행렬의 크기 3 * 3에 해당하는 정보 “3 3”을 입력하고, 해당하는 3*3 행렬을 입력받는다. 행렬의 입력이 끝나면 행렬을 행우선과 열우선으로 출력한다.

그리고 행렬 b에 대해서도 같은 정보를 입력받는다. 이 두 행렬을 입력받으면 바로 $c=a+b$ 연산을 수행하고 행렬 c를 출력한다.

```
sanghwan@sanghwan-VirtualBox: ~/dbox/classes191/ds/hw/hw192
File Edit View Search Terminal Help
sanghwan@sanghwan-VirtualBox:~/dbox/classes191/ds/hw/hw192$ java LabTest

Initialize a sparse matrix a
Enter number of rows and columns
3 3
0 2 0
1 0 5
3 2 0

Sparse Matrix a, Row major order
0 2 0
1 0 5
3 2 0

Sparse Matrix a, Column major order
0 1 3
2 0 2
0 5 0

Initialize a sparse matrix b
Enter number of rows and columns
3 3
1 0 0
1 0 0
2 4 0

Sparse Matrix b, Row major order
1 0 0
1 0 0
2 4 0

Sparse Matrix b, Column major order
1 1 2
0 0 4
0 0 0

Sparse Matrix c = a + b, Row major order
1 2 0
2 0 5
5 6 0

Sparse Matrix c = a + b, Column major order
1 2 5
2 0 6
0 5 0

Enter new value of an element (r, c, v)

```

```
sanghwan@sanghwan-VirtualBox: ~/dbox/classes191/ds/hw/hw192
File Edit View Search Terminal Help
sanghwan@sanghwan-VirtualBox:~/dbox/classes191/ds/hw/hw192$ java LabTest
Initialize a sparse matrix a
Enter number of rows and columns
3 3
0 2 0
1 0 5
3 2 0

Sparse Matrix a, Row major order
0 2 0
1 0 5
3 2 0

Sparse Matrix a, Column major order
0 1 3
2 0 2
0 5 0
Initialize a sparse matrix b
Enter number of rows and columns
3 3
1 0 0
1 0 0
2 4 0

Sparse Matrix b, Row major order
1 0 0
1 0 0
2 4 0

Sparse Matrix b, Column major order
1 1 2
0 0 4
0 0 0

Sparse Matrix c = a + b, Row major order
1 2 0
2 0 5
5 6 0

Sparse Matrix c = a + b, Column major order
1 2 5
2 0 6
0 5 0

Enter new value of an element (r, c, v)

```

더해진 행렬 c 가 만들어지면, 이 c 행렬의 특정 원소의 값을 바꾼다. 아래 그림에서 첫번째 예를 보면, “1 1 8” 값이 입력이 되는데, (1, 1)의 원소를 8로 바꾼다는 의미이다. 변경시키고 나서 다시 행렬 C 를 row major order와 column major order로 보여준다. (1,1)의 원소가 1에서 8로 변경되었음을 알 수 있다.

```
sanghwan@sanghwan-VirtualBox: ~/dbox/classes191/ds/hw/hw192
File Edit View Search Terminal Help

Sparse Matrix c = a + b, Row major order
1 2 0
2 0 5
5 6 0

Sparse Matrix c = a + b, Column major order
1 2 5
2 0 6
0 5 0

Enter new value of an element (r, c, v)
1 1 8

new value: 1 1 8

Sparse Matrix new C, Row major Order
8 2 0
2 0 5
5 6 0

Sparse Matrix new C, Column major Order
8 2 5
2 0 6
0 5 0

Enter new value of an element (r, c, v)
1 2 0

new value: 1 2 0

Sparse Matrix new C, Row major Order
8 0 0
2 0 5
5 6 0

Sparse Matrix new C, Column major Order
8 2 5
0 0 6
0 5 0

Enter new value of an element (r, c, v)

```

위와 같은 변경은 아래에서 정의된 Delete() 함수를 호출하고, 이후에 AddOneElement() 함수를 호출하는 방식으로 구현된다.

이 프로그램을 완성하기 위해서는 hw2.java에 정의된 함수들을 완성하면 된다.

- `boolean AddOneElement(final int r, final int c, final int v) :`
이 함수는 `SparseMatrix`에서 (r, c) 원소의 값으로 v 를 가진 새로운 원소를 추가하는 함수이다. 만약 (r, c) 원소가 기존에 존재하면 (즉 0이 아닌 값을 가지면) 행렬에 원소를 추가하지 않고, `false`를 `return` 하고, 그렇지 않으면 원소를 추가하고 `true`를 `return` 한다. 다만 v 가 0이면 추가하지 않는다.
- `boolean Delete(final int r, final int c):` 이 함수는 (r, c) 원소를 삭제하는 함수이다. 원소가 존재하면 그 원소를 삭제하고 `true`를 리턴하고, 존재하지 않으면 `false`를 리턴한다.
- `SparseMatrix Add(SparseMatrix b):` 이 함수는 자기 자신과 b 를 더하여 새로운 `SparseMatrix c`를 만들어 이를 `return` 하는 함수이다. `Polynomial`의 덧셈을 생각하면 쉽게 구현할 수 있다.

프로그램 테스트

컴파일

```
$ javac LinkedMatrix.java hw2.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
```

또는

```
$ diff -i --strip-trailing-cr -w abc lab.out
```

주의 사항

위의 프로그램 테스트에 실패한 파일은 제출하면 안됨.