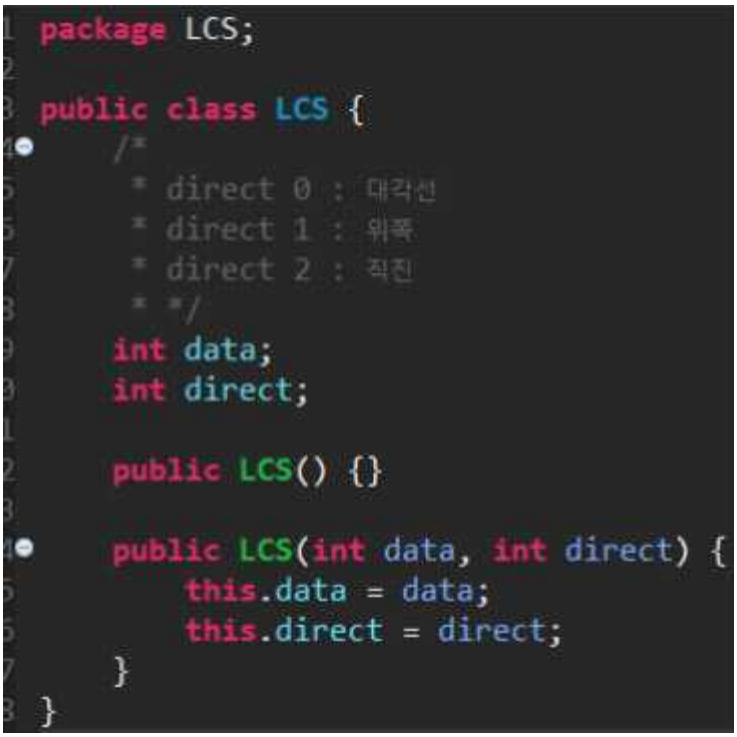
알고리즘

- 08. LCS -

제 출 일	2018.11.30
학 과	컴퓨터공학과
학 번	201402391
이 름	이 병 만

※ 과제 목표 및 해결 방법

※ 주요 부분 코드 설명 (알고리즘 부분 코드 캡쳐)



```
package LCS;

public class LCS {
    /*
     * direct 0 : 대각선
     * direct 1 : 위쪽
     * direct 2 : 직진
     * */
    int data;
    int direct;

    public LCS() {}

    public LCS(int data, int direct) {
        this.data = data;
        this.direct = direct;
    }
}
```

➔ LCS를 위한 클래스를 생성하였다. 해당 클래스에는 data와 방향을 저장하는 direct를 변수로 가지고 두 값을 저장할 수 있는 생성자를 가진다.

```java
public class File_read {
    public static ArrayList<String> fileRead(String filename) throws IOException {
        String path = main.class.getResource("").getPath(); // Get a Absolute path
        File read_file = new File(path + filename);
        // Create inputStream
        FileReader filereader = new FileReader(read_file);
        BufferedReader bufReader = new BufferedReader(filereader);
        String line = "";
        String[] a = null;
        ArrayList<String> arr = new ArrayList<String>();

        while ((line = bufReader.readLine()) != null) {
            a = line.split("\n");
            if (!isInteger(a[0]))
                arr.add(a[0]);
        }
        bufReader.close();
        return arr;
    }

    public static boolean isInteger(String s) {
        try {
            Integer.parseInt(s);
        } catch(NumberFormatException e) {
            return false;
        } catch(NullPointerException e) {
            return false;
        }
        return true;
    }
}
```

➔ txt 파일을 읽는 함수이다. 파일 이름을 인자로 받고 해당 파일이 있으면 한 줄씩 읽는다. 다 읽을 때 까지 반복한다. 읽은 문자가 숫자가 아니면 생략하고 문자열이면 배열에 문자열을 저장하고 파일을 다 읽으면 배열을 반환한다.

```java
public class File_write {
    public static void fileSave(String path, String str) {
        try {
            String savePath = main.class.getResource("").getPath(); // Get a Absolute path
            File file = new File(savePath + path);
            FileWriter fw = new FileWriter(file);
            BufferedWriter bufWriter = new BufferedWriter(fw);

            bufWriter.write(str);

            bufWriter.close();
            System.out.println(path + " saved");
        } catch(Exception e) {
            System.out.println(e);
        }
    }
}
```

➔ LCS를 구한 문자열을 txt파일에 저장하는 함수이다. 파일의 이름과 저장할 문자열을 받고 파일을 열어 해당 문자열을 쓰고 파일을 닫는다. 저장이 완료되면 파일명과 함께 저장되었다고 출력해준다.

```java
public static void lsc_length(String X, String Y) {
    /* variable */
    int m = X.length();
    int n = Y.length();
    String X_arr[] = new String[m+1];
    String Y_arr[] = new String[n+1];
    LCS c[][] = new LCS[m+1][n+1];

    /* create array */
    for (int i=0; i<X_arr.length-1; i++)
        X_arr[i+1] = X.substring(i, i+1);
    for (int i=0; i<Y_arr.length-1; i++)
        Y_arr[i+1] = Y.substring(i, i+1);

    /* c array setting */
    for (int i=1; i<=m; i++)
        c[i][0] = new LCS();
    for (int j=1; j<=n; j++)
        c[0][j] = new LCS();

    /* string check */
    for (int i=1; i<=m; i++) {
        for (int j=1; j<=n; j++) {
            if (X_arr[i].equals(Y_arr[j]))
                c[i][j] = new LCS(c[i-1][j-1].data + 1, 0);
            else if (c[i-1][j].data >= c[i][j-1].data)
                c[i][j] = new LCS(c[i-1][j].data, 1);
            else
                c[i][j] = new LCS(c[i][j-1].data, 2);
        }
    }
    for (int i=1; i<=m; i++) {
        System.out.println();
        for (int j=1; j<=n; j++)
            System.out.print("["+c[i][j].data +","+ c[i][j].direct +"] ");
    }
    System.out.println();
    File_write.fileSave("Output.txt", print_lcs(c, X_arr, m, n));
    str = ""; // 초기화
}
```

➔ 인자로 받은 문자열의 길이를 저장한다, 문자열을 하나씩 저장하기 위해서 배열을 생성한다. 데이터와 방향을 저장할 배열을 c를 생성한다. 초기값을 세팅하고 반복문을 실행하면서 두 문자열을 비교해서 같으면 현재있는 값에 1을 더한 값과 방향(대각선)을 함께 배열에 저장한다. 만약 오른쪽 아래의 값을 비교했을 때 왼쪽의 값이 더 크면 왼쪽의 값과 방향(위쪽)을 저장하고 그렇지 않으면 아래쪽 값과 방향(직직)을 저장한다. 아래 반복문은 출력을 위한 구문이다. 파일을 저장하는 함수를 호출하는데 인자로는 저장할 파일 이름과 LCS출력함수를 인자로 넣어준다.

```java
static String str = "";
public static String print_lcs(LCS[][] b, String[] X_arr, int i, int j) {
    if (i == 0 || j == 0)
            return str;
    if (b[i][j].direct == 0) {
        print_lcs(b, X_arr, i-1, j-1);
        return str += X_arr[i];
    }
    else if (b[i][j].direct == 1) {
        print_lcs(b, X_arr, i-1, j);
    }
    else {
        print_lcs(b, X_arr, i, j-1);
    }
    return str;
}
```

➔ LCS를 출력하는 함수이다. 재귀함수로 실행되며 방향이 0이면 대각선으로 이동하고 1이면 위로 올라가고 2이면 직진으로 이동하면서 I = 0 또는 j = 0 이면 함수를 종료한다. 이 때 재귀로 실행되었기 때문에 함수를 반환하면서 X_arr에 있는 값을 str에 더해주면서 LCS를 출력하였다.

※ 실행 결과 및 분석