

알고리즘 5주차

201902694 - 박재우

RANK	TEAM	SCORE	1-SORTING  [1 POINT]	2-PRICECOMPARISON  [3 POINTS]	3-FILEMANAGER  [6 POINTS]
13	201902694	10 298	26 2 tries	77 1 try	135 3 tries

Submissions			
time	problem	lang	result
10/05/20-16:27	3-FILEMANAGER	JAVA	CORRECT
10/05/20-16:06	3-FILEMANAGER	JAVA	WRONG-ANSWER
10/05/20-16:00	3-FILEMANAGER	JAVA	WRONG-ANSWER
10/05/20-15:28	2-PRICECOMPARISON	JAVA	CORRECT
10/05/20-14:37	1-SORTING	JAVA	CORRECT
10/05/20-14:25	1-SORTING	JAVA	RUN-ERROR

Clarifications
No clarifications.

Clarification Requests
No clarification request.

request clarification

문제 1

a. 문제 / 목표

입력받은 문자열 오름차순 정렬

b. 해결방법 (소스코드 첨부)

```
import java.util.ArrayList;

public class first {

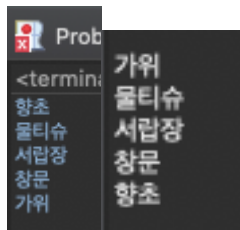
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        ArrayList<String> ar = new ArrayList<>();
        String temp;
        while(sc.hasNextLine()) {
            temp = sc.nextLine();
            if(temp.isEmpty())
                break;
            ar.add(temp);
        }
        Collections.sort(ar);

        Iterator iter = ar.iterator();
        while(iter.hasNext()) {
            System.out.println(iter.next());
        }
    }
}
```

Eof를 통해 문자열을 원할때까지 받으며 받은 것은 arraylist에 저장한다.

그후 입력이 전부 끝나면 collections sort를 이용해서 정렬한다.

c. 결과 (입력, 출력결과)



자신만의 생각 / 느낀점 :

처음에 eof에 대해서 생각이 나지 않았던점을 제외하면
쉬운 분제였다고 생각한다.

난이도 2/5

문제 2

a. 문제 / 목표

Ssd 스펙에 따라 용량 내림차순->응답속도오름차순->가격->이름 오름
차순 순서로 정렬한다

b. 해결방법 (소스코드 첨부)

```
public class second {  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        ArrayList<ssd> ar = new ArrayList<>();  
        String temp;  
        while(sc.hasNextLine()) {  
            temp = sc.nextLine();  
            if(temp.isEmpty())  
                break;  
            ssd t = new ssd(temp);  
            ar.add(t);  
        }  
        //size;  
        for(int i=0;i<ar.size()-1;i++) {  
            for(int j = 0;j<ar.size()-1-i;j++) {  
                if(ar.get(j).size < ar.get(j+1).size) {  
                    ssd tp = ar.get(j);  
                    ar.add(j,ar.get(j+1));  
                    ar.remove(j+1);  
                    ar.add(j+1,tp);  
                    ar.remove(j+2);  
                }  
            }  
        }  
        //speed  
        for(int i=0;i<ar.size()-1;i++) {  
            for(int j = 0;j<ar.size()-1-i;j++) {  
                if(ar.get(j).speed.compareTo(ar.get(j+1).speed) > 0 && ar.get(j).size <= ar.get(j+1).size) {  
                    ssd tp = ar.get(j);  
                    ar.add(j,ar.get(j+1));  
                    ar.remove(j+1);  
                    ar.add(j+1,tp);  
                    ar.remove(j+2);  
                }  
            }  
        }  
    }  
}
```

```

//price
for(int i=0;i<ar.size()-1;i++) {
    for(int j = 0;j<ar.size()-i-1;j++) {
        if(ar.get(j).price>ar.get(j+1).price && ar.get(j).size <= ar.get(j+1).size
            && ar.get(j).speed.compareTo(ar.get(j+1).speed) >= 0) {
            ssd tp = ar.get(j);
            ar.add(j,ar.get(j+1));
            ar.remove(j+1);
            ar.add(j+1,tp);
            ar.remove(j+2);
        }
    }
}

//name
for(int i=0;i<ar.size()-1;i++) {
    for(int j = 0;j<ar.size()-i-1;j++) {
        if(ar.get(j).name.compareTo(ar.get(j+1).name) > 0 && ar.get(j).price>=ar.get(j+1).price
            && ar.get(j).size <= ar.get(j+1).size
            && ar.get(j).speed.compareTo(ar.get(j+1).speed) >= 0) {
            ssd tp = ar.get(j);
            ar.add(j,ar.get(j+1));
            ar.remove(j+1);
            ar.add(j+1,tp);
            ar.remove(j+2);
        }
    }
}

for(int k=0;k<ar.size();k++)
    System.out.println(ar.get(k).name);
}

}

class ssd {
    String name;
    int size;
    String speed;
    int price;

    public ssd(String str) {
        String[] temp = str.split(" ");
        name = temp[0];
        size = Integer.parseInt(temp[1]);
        speed = temp[2];
        price = Integer.parseInt(temp[3]);
    }
}

```

Ssd객체를 생성하여 생성자를 통해 입력받은 문자열에서

이름,가격,속도,용량을 분리해서 저장한다.

각객체에 따라 용량, 속도, 가격 이름순으로 정렬하되

앞선 규칙에는 위반되지않게 정렬한다.

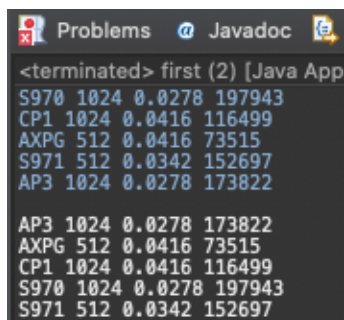
예를들어 용량에 따라 정렬했으면 속도 오름차순으로 정렬할때

용량의 내림차순 규칙을 어기지 않게 정렬한다.

For문을 통하여 비교를 통해 버블정렬로 구현했다.

앞선규칙은 그대로 가져와 순서대로 버블정렬을 응용한 정렬로 모든
규칙에 따라 정렬한다.

c. 결과 (입력, 출력결과)



```

<terminated> first (2) [Java App]
S970 1024 0.0278 197943
CP1 1024 0.0416 116499
AXPG 512 0.0416 73515
S971 512 0.0342 152697
AP3 1024 0.0278 173822

AP3 1024 0.0278 173822
AXPG 512 0.0416 73515
CP1 1024 0.0416 116499
S970 1024 0.0278 197943
S971 512 0.0342 152697

```

자신만의 생각 / 느낀점

2차원배열을 통해서도 풀 수 있다고들 했지만 나는 그 방식으로

풀 방법이 생각나지 않아 버블정렬을 이용해 정렬하는 조건을

문제의 조건에 맞추어서 총 4번의 정렬을 시도했다.

너무 노가다로 푼것같아 다른 풀이도 시도해볼 생각이다.

난이도 3/5

문제 3

a. 문제 / 목표

파일들을 확장자->파일이름->번호 순서로 각각 오름차순 규칙 정렬

b. 해결방법 (소스코드 첨부)

```
public class third {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        ArrayList<file> ar = new ArrayList<>();  
        String temp;  
        while(sc.hasNextLine()) {  
            temp = sc.nextLine();  
            if(temp.isEmpty())  
                break;  
            file f = new file(temp);  
            ar.add(f);  
        }  
        //format  
        for(int i=0;i<ar.size();i++) {  
            for(int j = 0;j<ar.size()-i-1;j++) {  
                if(ar.get(j).format.toLowerCase().compareTo(ar.get(j+1).format.toLowerCase()) > 0) {  
                    file tp = ar.get(j);  
                    ar.add(j,ar.get(j+1));  
                    ar.remove(j+1);  
                    ar.add(j+1,tp);  
                    ar.remove(j+2);  
                }  
            }  
        }  
        //name  
        for(int i=0;i<ar.size();i++) {  
            for(int j = 0;j<ar.size()-i-1;j++) {  
                if(ar.get(j).name.compareTo(ar.get(j+1).name) > 0 && ar.get(j).format.toLowerCase().compareTo(ar.get(j+1).format.toLowerCase()) >= 0) {  
                    file tp = ar.get(j);  
                    ar.add(j,ar.get(j+1));  
                    ar.remove(j+1);  
                    ar.add(j+1,tp);  
                    ar.remove(j+2);  
                }  
            }  
        }  
        for(int i=0;i<ar.size();i++) {  
            for(int j = 0;j<ar.size()-i-1;j++) {  
                if(ar.get(j).number > ar.get(j+1).number  
                && ar.get(j).name.compareTo(ar.get(j+1).name) >= 0 && ar.get(j).format.toLowerCase().compareTo(ar.get(j+1).format.toLowerCase()) >= 0) {  
                    file tp = ar.get(j);  
                    ar.add(j,ar.get(j+1));  
                    ar.remove(j+1);  
                    ar.add(j+1,tp);  
                    ar.remove(j+2);  
                }  
            }  
        }  
        for(int k=0;k<ar.size();k++)  
            System.out.println(ar.get(k).strs);  
    }  
}  
  
class file {  
    String format;  
    String name;  
    int number;  
    String strs;  
  
    public file(String str) {  
        strs = str;  
        String[] temp = str.split("[.]");  
        name = temp[0];  
        format = temp[1];  
        temp = name.split("-");  
        name = temp[0];  
        if(temp.length != 1)  
            number = Integer.parseInt(temp[1]);  
        else  
            number = 0;  
    }  
}
```

앞선 방법과 똑같이 file객체를 생성해 입력받은 문자열을 문자열 그대로 저장하고 그 문자열을 확장자,이름,숫자로 분리해서 저장한다. 그후 버블정렬을 응용해서 문자열 비교를 통해 확장자 순서대로 정렬한다. 그후 확장자 정렬조건을 해치지 않은 채 이름순으로 문자열 비교와 버블정렬 응용을 통해서 정렬한다.

이후 마찬가지로 앞선 정렬규칙을 해치지 않는 규칙을 적용한후 파일 숫자 오름차순 순으로 버블정렬응용을 통해서 정렬한다.

김완재

c. 결과 (입력, 출력결과)

자신만의 생각 / 느낀점 :

실습시간에 객체생성과 정규식 표현으로 문제를 풀려고 노력하러 했지만 정규식 표현에 대해서 생각이 나지 않아 정규식으로는 제대로 풀기 어려웠다. 그래서 앞의 2번문제처럼 단순한 생각으로 문제를 풀었다. 정규식에 대해서 따로 공부를 해야할것 같다.

난이도 5/5