Relational MapReduce Patterns #3 - Difference

HackerRank

Mappers and Reducers

Here's a quick but comprehensive introduction to the idea of splitting tasks into a MapReduce model. The four important functions involved are:

```
Map (the mapper function)
EmitIntermediate(the intermediate key,value pairs emitted by the mapper functions)
Reduce (the reducer function)
Emit (the final output, after summarization from the Reduce functions)
```

We provide you with a single system, single thread version of a basic MapReduce implementation.

Task

Given two sets of integers, **R** and **S** select and display the difference, **R** - **S**. The relative ordering of integers in the intersecting set should be same as that in set **R**.

The code for the MapReduce class, reading and splitting the text, parts related to IO etc. has already been provided. However, for this particlar task, you ONLY need parts of the mapper and its related functions. However certain parts of the mapper and reducer functions are incomplete. You need to replace the questionmarks (?). Your task is to fill up these question marks appropriately, such that the program works, and accomplishes the specified task. Also, this program may output certain information to the error stream. This information has been logged to help beginners gain a better understanding of the the intermediate steps in a map-reduce process.

Language Support

Java and Ruby.

Input Format

The first line contains two space separated integers **Nr** and **Ns**, which are the number of elements in the set **R** and **S** respectively. This is followed by **Nr** integers, the elements of set **R**, each on a new line, such that -100 <= **X** <= 100. This is followed by **Ns** integers, the elements of set **S**, each on a new line, such that -100 <= **X** <= 100. Also, 10 <= **Nr**,*Ns* <= 100
For instance, if **R** = [10,20,40,20,60];you may treat it as [10,20,40,60]

Output Format

Output each of the integers in the difference set, **R** - **S**. The relative ordering of integers in the intersecting set should be same as that in set **R**.

Sample Input

```
10 10
-51
-43
74
-96
24
-14
11
77
-45
-90
45
8
29
0
-43
-13
-72
71
-96
-26
```

Sample Output

```
-51
74
24
-14
11
77
-45
-90
```

Explanation

We have two arrays with 10 elements each. We list out only the integers which are present in **R** but not present in **S**. We display them in the same order as they existed in **R** (i.e, the first set).