Specification of 2.3(4)

In this homework, you need to design the data structure and implement the five actions described in *hw2.pdf*. Please be aware that the dataset is super-huge, about 2G. Thus, if you are not careful with your implementations, your program can easily crash. Also, **DO NOT copy the dataset to your own directory**. Your own directory is on the NFS system and copying it there would only slow down your program (and other users' programs).

The input/output formats are in the following sections. There is no sample output this time and you'll need to justify your code by yourself.

Dataset Clarification

- The path of dataset is always `/tmp2/KDDCUP2012/track1/rec_log_train.txt`, so you can simply write this path in your code.
- The TAs will use the same dataset with different queries when testing your program, so if you find some useful features in this dataset, feel free to take advantage of them.
- There may be duplicate log record in the dataset (i.e. exists two different lines' (UserId, ItemId, Result, Unix-timestamp) are all equivalent). In this case, you have to remove duplicate log record (or the answer may not be correct).

Query Clarification

- ratio($i, threshold$)=(#accept)/(#total): "#total" means the number of users who are given items by the system for MORE THAN *threshold* times (Any item recommendation, i.e. not only recommendation of Itemid=i but also other items, regardless of accept or reject). "#accept" means the number of users among #total who have accepted Itemid=i. Note that "MORE THAN *threshold*" means $> threshold$.

Input Format

The first line is the number n, representing the number of testing actions ($n \leq 2000$). Each testing action contains two lines, the first one is the action name, *accept, items, users, ratio, findtime_item*. And the second line contains the parameters:

- accept(u,i,t): 3 integers separated by a space, u i t.
- items(u1,u2): 2 integers separated by a space, $u1$ $u2$.
- users(i1,i2,t1,t2): 4 integers separated by a space, $i1$ $i2$ $t1$ $t2$, with $t1 \leq t2$.
- ratio(i,threshold): 2 integers separated by a space, i *threshold*.
- findtime_item(i,Us): the first integer is i and the rest are members of *Us*, separated by spaces. There will be at lease one user in *Us*.

The TAs will only test your program with valid (ItemId), (UserId), (Unix-timestamp) so there is no serious need of error handling in this part.

Output Format

For each action, follow the directions below for outputs.

- accept(u,i,t): outputs one line, 1 for acceptance, -1 for rejection and 0 for no record.
- items(u1,u2): outputs the sorted (ItemId) line by line in ascending order (remove duplication).
- users(i1,i2,t1,t2): outputs the sorted (UserId) line by line in ascending order (remove duplication).

- ratio(i,threshold): outputs one line in (#accept)/(#total) format, like 14/101. Please DO NOT reduce the fraction and just output the raw numbers.
- findtime_item(i,Us): outputs the sorted (Unix-timestamp) line by line in ascending order (remove duplication).

For actions *items*, *users* and *findtime_time*, if the output list is empty, please print a string "EMPTY" (without quotes) in a newline.

Sample Input

```
5
accept
494303 324861 1321027198
items
460266 463359
users
514413 324861 1318348790 1321027199
ratio
908591 1000
findtime_item
651131 2307074 617676 1853982 592712
```

Submission Guideline

Please submit your code with GitHub as directed in the GitHub submission guide/tutorial video. Your directory structure (under GitHub repository) should be:

- `hw2/*`, your source code.
- `hw2/Makefile`, where the TAs can use the command "`make`" on CSIE R217 linux machines to compile your code, and "`make run`" to run your program.
- `hw2/README.txt` (optional), anything you want the TAs to read before grading your code.

You have to make sure your code can be compiled on CSIE R217 linux machines with your `Makefile` and run normally since we will test your program on CSIE R217 linux machines.