# 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.

# Input Format

The first line is a 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.
- users(i1,i2,t1,t2): outputs the sorted (UserId) line by line in ascending order.
- ratio(i,threshold): outputs one line in (#accept)/(#total) format, like 14/101. There is no need to reduce the fraction and you can just use the raw numbers.
- findtime_item(i,Us): outputs the sorted (Unix-timestamp) line by line in ascending order.

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
```