

Specifications for Homework 2.4

In this spec, first we will tell you how the TAs are going to test your program. Next, the formats of the input and output will be given with full instructions and some examples. In the end, some common Q&A are listed.

Test

You should write a Makefile on your own. The TAs will use *make* to compile your source code and use the following command to test your program:

```
$ ./demo [file path]
```

For example,

```
$ ./demo /tmp2/KDDCup2012/track2/kddcup2012track2.txt
```

Note that the designated file path will be set by the TAs. Your program should open the designated file and read it.

Format

Your demo program should work like a simple command line. That is, after executing the demo file, it can let the user enter some commands. **Each command takes one line.** ~~For example/ ... (This part is not to be used by the TA to begin with/ each line)~~

And your program should read the command on **stdin** and give the response on **stdout**. Also, you have to separate your response with a line of 20 “*”. For example:

```
*****
# your output
# ...
# your output in the same respond
*****
```

In this homework, the TA will use 5 commands:

- Get: **get**(u, a, q, p, d)
- Clicked: **clicked**(u)
- Impressed: **impressed**($u1, u2$)
- Profit: **profit**(a, θ)
- Quit: User can use this command to leave the running program.

The formats for each command are listed below, and we will give an overall example at the end of this section. **(Please pay attention to the comments with ★)**

1) Get

- input:

```
get [u] [a] [q] [p] [d]
```

- output:

```
*****
[Click] [Impression]
*****
```

- ★ You should **sum up** the output (Click, Impression)'s at $M[UserID][AdID][QueryID][Position][Depth]$ instead of listing all of them.

2) Clicked

- input:

```
clicked [u]
```

- output:

```
*****
[AdID] [QueryID]
.
.
5 [AdID] [QueryID]
*****
```

- ★ No duplicated results.
- ★ Please sort the results with increasing AdID order first. If the AdIDs are the same, then sort with increasing QueryID order.

3) Impressed

- input:

```
impressed [u1] [u2]
```

- output:

```
*****
[AdID]
\t[URL] [AdvertiserID] [KeywordID] [TitleID] [DescriptionID]
.
5 .
\t[URL] [AdvertiserID] [KeywordID] [TitleID] [DescriptionID]
*****
```

- ★ Note that the `\t` means a **TAB**. Please sort the results with the AdID in an increasing order. Let **property** be a combination of “[URL] [AdvertiserID] [KeywordID] [TitleID] [DescriptionID]”. If there's an AdID having many different properties, you have to show all the different **properties that either user has an impression on**. However, it's OK that the properties are not sorted.

4) Profit

- input:

```
profit [a] [theta]
```

- output:

```
*****
[UserID]
.
.
5 [UserID]
*****
```

- ★ Please sort the results with the UserID in an increasing order.

5) Quit

- input:

```
quit
```

- output:

```
# leave the program
```

Example

In the following examples, we will use the sample track file: *demotrack.txt*. You can get the file [on the course website](#). Note that this is only for demonstrating the input/output format, the TA will not use this sample track file in the final testing.

- input:

```
clicked 490234
get 490234 21560664 2255103 2 2
profit 21375650 0.5
impressed 6231944 490234
5 quit
```

- output:

```

*****
21375650 4688625
21560664 2255103
*****
5 *****
1 2
*****
*****
490234
10 6231944
*****
*****
21375650
    10196385171799537224 36832 202465 457316 429545
15 21560664
    4860571499428580850 37484 317 48989 44771
    378261549181239 37484 333 56892 42311
*****
# leave the program

```

Q&A

- 1) Format meanings such as depth, position, query, and so on.
<http://www.kddcup2012.org/c/kddcup2012-track2>
- 2) What is the type of theta of profit(a,theta)
 Double
- 3) Should I read the whole file at one time or by part? Which is better for searching, sorting or computing?
 Read at once may be better
- 4) Should I use a temporary file for convenient access?
 Memory is large enough, so you don't need to use a temporary file.
- 5) How to design an efficient way for saving data?
 Find out the tasks that are done a lot, and make these tasks more efficient.
- 6) Are there any repeated data when TA testing our program? For example, once there are some repeated data in Get, what should I do?
 You should sum up the output(click, impression) in function Get
- 7) How to save the data(vector, deque, set, multimap, map)? How to search, sort and compute?
 We suggest you to construct some array or map in order to make some special index for different applications.

Final Remarks

If you still don't understand the instructions, please feel free to come to TA hours and ask for help!