Test assignment

Introduction

Here is a short snippet of a log file. The goal of the assignment is to create a program that parses the log file and extracts, aggregates and presents the data from it. The log file contains request durations. The file format is best explained with the following examples:

```
2015-08-19 00:00:01,049 (http--0.0.0.0-28080-405) [] /checkSession.do in 187
2015-08-19 00:00:01,963 (http--0.0.0.0-28080-245) [] /checkSession.do in 113
2015-08-19 00:00:02,814 (http--0.0.0.0-28080-245) [CUST:CUS5T27233]
/substypechange.do?msisdn=300501633574 in 17
2015-08-19 00:00:03,260 (http--0.0.0.0-28080-245) [CUST:CUS5T27233]
/mainContent.do?action=TOOLS&contentId=main_tools in 5
```

It contains:

- date
- timestamp
- thread-ID (in brackets)
- optional user context (in square brackets)
- URI + query string
- string “in”
- request duration in milliseconds

..and..

```
2015-08-19 00:04:45,212 (http--0.0.0.0-28080-405) [] updateSubscriptionFromBackend
300445599231 in 203
2015-08-19 00:04:45,259 (http--0.0.0.0-28080-405) [ASP CUST:CUS5T27233]
getPermission
300445599231 in 32
```

which contains:

- date
- timestamp
- thread-ID (in brackets)
- optional user context (in square brackets)
- requested resource name (one string)
- data payload elements for resource (0..n elements)
- string “in”
- request duration in milliseconds

What to do?

Required interface:

1. The program needs to run from command line and must take log filename as an argument.
2. The program must display help information documenting how to use it when running with **-h** argument

Sample of expected command line:

```
aks-mbp:dist ak$ java -jar assignment.jar timing.log 10
```

Functional requirements:

1. Print out top **n** (exact value of n is passed as program argument) resources with highest average request duration.
2. Draw histogram of hourly number of requests.
3. Print out number of (milli)seconds your program ran