Names: Bob Skowron, Jason Walker
Keys: rskowron, jwalker
SVN: jwalker: https://svn.seas.wustl.edu/repositories/jwalker/cse427s_fl17/

1. a. The data format is a flat text file where each record represents an HTTP protocol request served by the server generating the log file taking requests from hosts whose IP is recorded within the log record. The information stored in each record is:

      - IP Address
      - Timestamp (date and time)
      - URL accessed
      - HTTP response code
      - Elapsed time of the request

      The test log has 5000 rows and the full data set has 4,477,843, so our sample is approximately .11% of the total.

   b. Mapper Input:
      (byteoffset, 10.223.157.186 - - [15/Jul/2009:21:24:17 -0700] ”GET /assets/img/media.jpg HTTP/1.1” 200 110997)
      (byteoffset, 10.223.157.186 - - [15/Jul/2009:21:24:18 -0700] ”GET /assets/img/pdf-icon.gif HTTP/1.1” 200 228)
      (byteoffset, 10.216.113.172 - - [16/Jul/2009:02:51:28 -0700] ”GET / HTTP/1.1” 200 7616)
      (byteoffset, 10.216.113.172 - - [16/Jul/2009:02:51:29 -0700] ”GET /assets/js/lowpro.js HTTP/1 .1” 200 10469)
      (byteoffset, 10.216.113.172 - - [16/Jul/2009:02:51:29 -0700] ”GET /assets/css/reset.css HTTP/1.1” 200 1014)

      Mapper Output:
      (10.223.157.186, 1)
      (10.223.157.186, 1)
      (10.216.113.172, 1)
      (10.216.113.172, 1)
      (10.216.113.172, 1)

      Reducer Input:
      (10.223.157.186, [1, 1])
      (10.216.113.172, [1, 1, 1])

      Reducer Output:
      (10.223.157.186, 2)
      (10.216.113.172, 3)

      The reducer is summing over an array of ones, same as we saw with the word count examples.

   c. See SVN

   d. i. Running the job locally, your inputs and outputs are local files and folders, respectively. When running on the cluster, the inputs and outputs are stored in hdfs. Print outputs, when run locally, are written out to the console for the end user to review. When run on the cluster, those outputs are not written out to the user. Running locally, there is no job management. The current job is what is run. When run on the cluster, hadoop (namely YARN) has to manage the resources on the cluster.

      ii. *hadoop jar logfileanalysis.jar stubs.ProcessLogs -fs=file:/// -jt=local /workspace/log_file_analysis/src/test_log_file output_test*

      iii. Using Eclipse is a personal preference. The first reason is the JAR file does not need to be compiled and stored on the filesystem. Eclipse is smart enough to use the classes within the project currently under development. This reduces the chances of forgetting to recompile the JAR after editing. The second reason is that command line arguments and inputs can easily be passed to the Run Configuration without having to tell hadoop to use the local filesystem. Each configuration can then be edited or customized as needed without having to go to/from the command line.

   e. Based on the output, there are 10 unique IP addresses in the testlog file. Yes, every line contributed to a count. The sum of the counts in the output file sum to 5000, which is the original number of line items

   f. When running the job on an actual cluster we need to keep in mind:

      - Size of the input file we will be processing as well as the number of records.

- Number of reducers we are using to process the expected number of records.
- Communication cost for both Mapper and Reducer based on the numer of records.
- Compute resources, ex. slots, CPUs, RAM, etc. available to the cluster.
- Storage of both intermediate as well as final MR output on HDFS.

- Total number of IP Addresses: 333,923
- 10.1.100.199: 35
- 10.1.100.5: 1
- 10.99.99.58: 21

The results are globally sorted because there is only one reducer. The shuffle and sort passes all keys to the single reducer in sorted order.

Names: Bob Skowron, Jason Walker
Keys: rskowron, jwalker
SVN: jwalker: https://svn.seas.wustl.edu/repositories/jwalker/cse427s_fl17/

2. a. See SVN

   b. The communication cost of the job is given by:
   number of key-value pairs that are Mapper input (n) +
   number of key-list-of-values pairs that are Reducer input (m)
   Here, n + m = 173,126 + 305,680 = 478,806

   c. Simply measuring the co-occurrence words next to one another may not give the entire context of the text. For example:
   The most interesting bird is the falcon
   This class is very hard
   Ideally, we would like to identify (bird, falcon) or (interesting, falcon) together to gain any information. If we only focus on pairs we lose the context of the sentence. If we were trying to identify sentiment about falcons, we wouldn't be able to discern much. Similarly, when modifiers do not come directly before or after a word of interest, a pairs based co-occurrence falls short. Even in the latter example if we remove the stop words we would only get (class, very) and (very, hard). Hardly a useful set of word pairings to determine the sentiment of a student taking this class.

   d. Because the text is stored in HDFS, it may be the case that a line of text goes over a block boundary. Now, hadoop should handle this, and for side by side pairs this is not too much to ask. However, if you were to be looking at neighbors of words that were several steps away, hadoop would need to buffer potentially a lot of data to map all the pairs. This could be very memory intensive.