

The Map-Reduce Programming Paradigm

M.Sc. course on “Technologies for Big Data Analysis” - Assignment 2

CHRISTOS BALAKTSIS (506) and VASILEIOS PAPASTERGIOS (505), Aristotle University, Greece

1 Introduction

The current document is a technical report for the second programming assignment in the M.Sc. course on *Technologies for Big Data Analysis*, offered by the *DWS M.Sc Program*¹ of the Aristotle University of Thessaloniki, Greece. The course is taught by Professor Apostolos Papadopoulos². The authors attended the course during their first year of Ph.D. studies at the Institution.

The assignment contains 4 sub-problems and is part of a series, comprising 3 programming assignments on the following topics:

Assignment 1 Multi-threading Programming and Inter-Process Communication

Assignment 2 The Map-Reduce Programming Paradigm

Assignment 3 Big Data Analytics with Scala and Apache Spark

In this document we focus on Assignment 2 and its 4 sub-problems. We refer to them as *problems* in the rest of the document for simplicity. The source code of our solution has been made available at the following public repository in the GitHub platform: <https://github.com/Bilpapster/big-data-playground>.

Roadmap. The rest of our work is structured as follows. We devote one section to each one of the 4 problems. That means problems 1, 2, 3 and 4 are presented in section 2, section 3, section 4 and section 5 respectively. For each problem, we first provide the problem statement, as given by the assignment. Next, we thoroughly present the reasoning and/or methodology we have adopted to approach the problem and devise a solution. Wherever applicable, we also provide insights about the source code implementation we have developed. Finally, we conclude our work in section 6. The appendix includes the evaluation results for any issues that necessitated them.

2 Problem 1: Numeronyms

We discuss here the first problem of the assignment. The main target of the assignment is to get acquainted with the MapReduce programming model in Hadoop framework in Java programming language. This is a WordCount problem’s variation.

2.1 Problem Statement

Implement a **MapReduce** program, a variation of the **word-count** problem, to generate and count “numeronyms”. Numeronyms correspond to words as shown below:

- s5n – shorten
- h7k – hyperlink
- l10n – localization

¹<https://dws.csd.auth.gr/>

²<https://datalab-old.csd.auth.gr/~apostol/>

- i18n – internationalization

A numeronym of a word is defined as an alphanumeric string formed by taking the first and last character of the word and inserting between them the count of characters between the first and the last. More specifically, the program should process words with a length of 3 characters or more, generate numeronyms, and then output the frequency of each numeronym.

The program should ignore:

- words shorter than 3 characters
- punctuation marks
- uppercase/lowercase differences, i.e., it should be *case-insensitive*

Additionally, a parameter k will be given, representing the minimum number of occurrences of a numeronym that we are interested in. The program output should be a list of numeronyms and the frequency of each numeronym that is greater than or equal to the parameter k specified by the user.

2.2 Proposed approach

2.2.1 Setting. Our implementation is run and tested in a Linux environment with 12 cores, using the Java programming language. We have used the Java Development Kit (JDK) version 11.0.11. The source code is developed in IntelliJ IDEA Community Edition 2021.1.1 and managed using Maven as the build tool.

The project's dependencies, including Hadoop libraries, are defined in the `pom.xml` file located in the root of the repository. The project is compiled and executed directly from IntelliJ IDEA.

To run the project, open the `NumeronymsMaster` class in IntelliJ IDEA, and execute the main method. You will need to specify the following command-line arguments:

- `<input_path>` specifies the directory containing the input text files, located at `map-reduce/ numeronyms /input`.
- `<output_path>` specifies the directory where the MapReduce output will be written, which will be created in the `out` folder.
- `<k>` is the minimum frequency threshold for numeronyms to be included in the results.

IntelliJ IDEA will handle the compilation and execution automatically when the main method is run. Make sure to configure the input and output paths as required for your specific run.

Note that the command-line arguments must follow the specified order and format. If any of the arguments are missing or invalid, the program will terminate with an appropriate error message.

2.2.2 Implementation. The proposed approach leverages the MapReduce programming model to compute numeronyms from a given input dataset. A numeronym is a concise representation of a word, typically formed by retaining the first and last letters while replacing the intervening characters with their count. This section outlines the methodology implemented across three key components: the driver class, the mapper, and the reducer. This methodology supports flexible configuration through command-line arguments, enabling users to specify the threshold k and the minimum word length for numeronym generation. The use of the MapReduce paradigm ensures scalability and parallel processing for large datasets, making the approach well-suited for distributed systems.

The driver class, `NumeronymsMaster`, orchestrates the overall MapReduce job. It takes command-line arguments for the input and output directories as well as a threshold parameter k , which determines the minimum frequency for numeronyms to be included in the final output. The following steps are executed:

- Input and output paths are parsed, and any pre-existing output directory is deleted using the `Utilities.deleteDirectory` method.
- A Hadoop configuration object is initialized with two parameters:

- `numeronyms.k`: The threshold k .
- `numeronyms.l`: The minimum word length (default value is 3) required for numeronym generation.
- The job is configured to use the `NumeronymMapper` and `NumeronymReducer` classes, with input and output formats set to `TextInputFormat` and `TextOutputFormat`, respectively.
- Finally, the job waits for completion before terminating.

The `NumeronymMapper` class is responsible for processing each line of the input text file and emitting numeronym -frequency pairs. The following steps are performed:

- During setup, the minimum word length (`numeronyms.l`) is retrieved from the configuration.
- Each input line is tokenized into individual words, and non-alphabetic characters are removed from each token.
- For valid words (those with lengths greater than or equal to the specified minimum), a numeronym is constructed by concatenating the first character, the number of intervening characters, and the last character.
- A key-value pair is emitted, where the key is the numeronym and the value is a count of one.

The `NumeronymReducer` class aggregates the frequency counts for each numeronym and filters out those below the specified threshold k . The process includes:

- Retrieving the threshold value (`numeronyms.k`) from the configuration during the setup phase.
- Summing the values for each numeronym key received from the mapper.
- Emitting numeronyms that meet or exceed the threshold k along with their corresponding frequencies.

2.2.3 Evaluation. The experiments were executed using the dataset ‘`SherlockHolmes.txt`’ with a parameter setting of $k = 10$. This configuration was chosen to evaluate the performance and the behavior of the system under moderate conditions. The dataset, containing a series of textual records, was processed to analyze various patterns and results.

To provide a comprehensive overview, the results of the experiment, including the detailed values derived from the dataset, are presented in Appendix A in a 2-column format.

3 Problem 2: Movie Analytics

The second problem focuses on producing analytic insights from an IMDB dataset.

3.1 Problem Statement

Implement a **MapReduce** program to perform analytics tasks on an IMDB dataset about movies. The utter goal of your analysis would be to extract useful insights from the available movie data that will assist the IMDB team provide better recommendations for movies, based on their genre and/or country. In particular, the dataset (`movies.csv`) contains the following fields:

- `imdbID`: unique identifier of the movie in the IMDB database
- `title`: the movie title
- `year`: the year the movie was first released
- `duration`: the duration of the movie
- `genre(s)`: the genre or genres in which the movie is classified
- `premier date`: the date of the first showing of the movie
- `score`: the IMDB score of the movie
- `country/-ies`: the country or countries the movie was produced in

You are asked to implement Map-Reduce source code in Java programming language for the following analytics tasks:

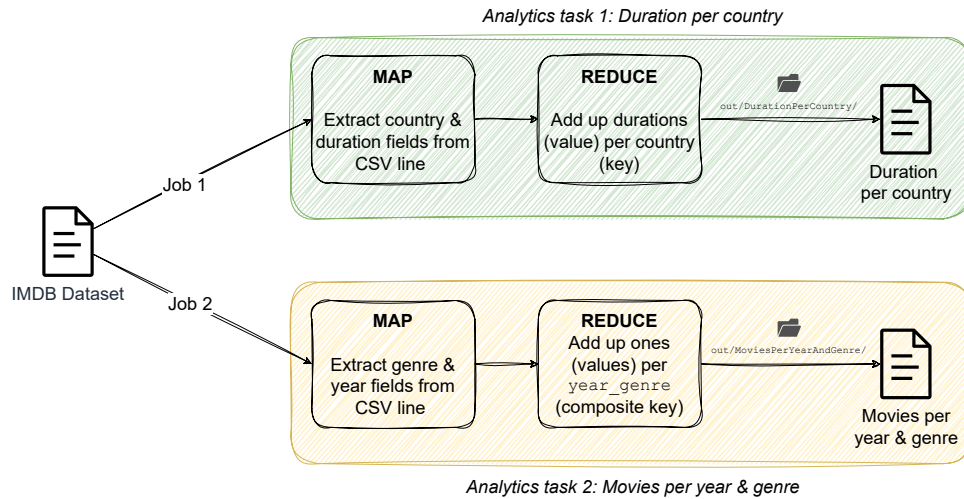


Fig. 1. The MapReduce solution architecture for the movie analytics problem. Two separate jobs are executed to solve the two tasks, namely duration per country (green) and movies per year & genre (yellow). The two tasks also create separate subdirectories for writing the results.

- Calculate the total duration of all movies per country. Note that in case multiple countries are recorded for a movie, the respective duration should be counted for all of them separately.
- Calculate the total number of movies per year and genre, having IMDB score over 8. For movies that have more than one genre, the sum should be separate for each genre.

3.2 Proposed approach

3.2.1 Setting. Our implementation is run and tested in a Linux environment with 12 cores, using the Java programming language. We have used the Java Development Kit (JDK) version 11.0.11. The source code is developed in IntelliJ IDEA Community Edition 2021.1.1 and managed using Maven as the build tool.

The project's dependencies, including Hadoop libraries, are defined in the `pom.xml` file located in the root of the repository. The project is compiled and executed directly from IntelliJ IDEA.

To run the project, open the `MovieAnalyticsMaster` class in IntelliJ IDEA, and execute the main method. You will need to specify the following command-line arguments:

- `<input_path>` specifies the directory containing the input text files, located at `map-reduce/movieAnalytics/input`.
- `<output_path>` specifies the directory where the MapReduce output will be written, which will be created in the `out` folder.

IntelliJ IDEA will handle the compilation and execution automatically when the main method is run. Make sure to configure the input and output paths as required for your specific run. Note that the command-line arguments must follow the specified order and format. If any of the arguments are missing or invalid, the program will terminate with an appropriate error message.

3.2.2 Implementation. The proposed approach leverages the MapReduce programming paradigm to compute the analytical insights for the two tasks. Figure 1 depicts the architecture of our solution as a diagram. We opt for executing two separate jobs; one for each task presented by the problem statement.

Task 1: Duration per country. One map-reduce cycle is enough to handle this task, as depicted in the top part of Figure 1 in green color. The map function parses the CSV file line by line and extracts the useful fields from each line. In this case, the useful fields are the country (or countries) the movie was produced and the movie duration, i.e., the fourth and ninth fields in the input line respectively. We employ more complex logic to handle cases where there are multiple countries in a single movie. In particular, we parse again the field and tokenize it into the separate countries, producing a key-value pair for each country-duration pair within a movie. The reduce function is, then, trivial; it just adds up the durations (value) per country (key) and outputs the results. The interested reader can refer to the `CSVProcessor.java` (mapper), `AnayticsEngine.java` (reducer) and `MovieAnalyticsMaster.java` (driver) files for the source code implementation of our solution.

Task 2: Movies per year & genre w.r.t. score constraint. The task is similar to the first one, with the only difference lying in the fields extracted from the input CSV line. In this case, we are interested about the year, the genre (or genres) and score fields, i.e., the third, fifth and seventh fields in the input CSV line. The map function produces key-value pairs in the form `(composite_key, 1)`, where the composite key consists of the year and the genre of the respective movie concatenated by an underscore, e.g., `2024_action`. We handle multiple genres per movie similarly with the multiple countries in task 1. We employ the same trivial reducer that adds up the values (ones) per (composite) key, as shown in the bottom part of Figure 1 in yellow color.

3.2.3 Evaluation. Our solution is tested using the provided IMDB dataset, namely `movies.csv`. We list the execution results in Appendix B.

4 Problem 3: DNA Sequence Patterns

The third problem focuses on mining unstructured data, namely DNA sequences, encoded in text format.

4.1 Problem Statement

A DNA sequence consists of four distinct symbols, namely A, G, C, and T. For instance, the following lines represent a part of the DNA sequence of the E. Coli bacterium ³:

```
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAGAGTGTCTGATAGCAGC
TTCTGAAGTGGTTACCTGCCGTGAGTAAATTTAAATTTTATTGACTTAGGTCACTAAATACTTTAACCAA
TATAGGCATAGCGCACAGACAGATAAAAATTACAGAGTACACAACATCCATGAAACGCATTAGCACCACC
ATTACCACCACCATCACCATTACCACAGGTAACGGTGCGGGCTGACGCGTACAGGAAACACAGAAAAAAG
```

Implement a **MapReduce** program that computes the frequency (i.e., number of occurrences) of all subsequences of lengths 2, 3, and 4 that are present in the input DNA sequence. Note that the processing of each line should be independent of all other lines. Use the file `ecoli.txt` to test your solution.

4.2 Proposed approach

4.2.1 Setting. Our implementation is run and tested in a Linux environment with 12 cores, using the Java programming language. We have used the Java Development Kit (JDK) version 11.0.11. The source code is developed in IntelliJ IDEA Community Edition 2021.1.1 and managed using Maven as the build tool.

The project's dependencies, including Hadoop libraries, are defined in the `pom.xml` file located in the root of the repository. The project is compiled and executed directly from IntelliJ IDEA. To run the project, open the `DNASequenceAnalyticsMaster` class in IntelliJ IDEA, and execute the main method. You will need to specify the following command-line arguments:

³https://en.wikipedia.org/wiki/Pathogenic_Escherichia_coli

Algorithm 1 Populating DNA subsequences

```

1: ▷ Input: a DNA sequence line (variable line)
2: for  $length \in \{2, 3, 4\}$  do
3:   Create a sliding window window of length length.
4:   Place left end of window at the start of line.
5:   while right end of window has not exceeded line do
6:      $subsequence \leftarrow$  contents of window
7:     Yield (subsequence, 1) key-value pair.
8:   Slide window to the right by 1.

```

- `<input_path>` specifies the directory containing the input text files, located at `map-reduce/dnaSequencePatterns/input`.
- `<output_path>` specifies the directory where the MapReduce output will be written, which will be created in the out folder.

IntelliJ IDEA will handle the compilation and execution automatically when the main method is run. Make sure to configure the input and output paths as required for your specific run. Note that the command-line arguments must follow the specified order and format. If any of the arguments are missing or invalid, the program will terminate with an appropriate error message.

4.2.2 Implementation. Algorithm 1 summarizes the most substantial part of our implementation, namely the logic that is implemented inside the mappers of our solution. At every execution of the map function, a DNA sequence line is given as input to the mapper. The latter is responsible for populating all possible subsequences of length 2, 3 and 4. We employ a sliding window that is nested inside a for-loop defining the subsequence length. The source code implementation of Algorithm 1 can be found in the `DNASequenceProcessor.java` file.

4.2.3 Evaluation. Our solution is tested using the provided DNA sequence, namely `ecoli.txt`. We list the execution results in Appendix C.

5 Problem 4: Probabilistic graph

We discuss here the fourth problem of the assignment. The main target of the assignment is to get acquainted with performing various MapReduce phases on a pipeline frame.

5.1 Problem Statement

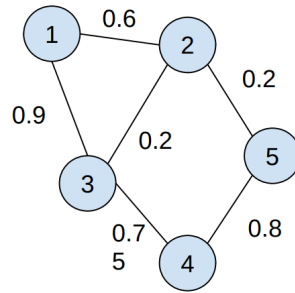
We are given an input file in text format where each line contains a connection between two vertices of a network and a probability value. For each edge e , there is a probability value $p(e)$ which indicates the probability that the two vertices are connected by the edge. Obviously, the values of $p(e)$ range between 0 and 1. The values in each line are separated by a space. Consider the network shown in the previous figure. The edge connecting vertices 4 and 5 has a probability of 0.8, the edge connecting vertices 2 and 3 has a probability of 0.2, etc.

The file corresponding to this graph would be:

1 2 0.6, 1 3 0.9, 2 3 0.2, 3 4 0.75, 2 5 0.2, 4 5 0.8,

The edges are generally stored in random order in the file, so we cannot assume they have a specific arrangement. The following tasks are requested:

- (1) Write a Java program that computes the average degree for all the vertices.
- (2) The average degree is defined as the sum of the probabilities of the edges that fall on a vertex.



- (3) For example, in the previous diagram, the average degree of vertex 3 is $0.9 + 0.2 + 0.75 = 1.85$.
- (4) Before performing this calculation, you should ignore all edges with a probability less than a threshold T , which should be passed as a parameter to the main function.
- (5) Modify the code from task 1 so that, at the end, only the vertices with an average degree greater than the average of the degrees of all the vertices are displayed in the output.

5.2 Proposed approach

5.2.1 Setting. Our implementation is run and tested in a Linux environment with 12 cores, using the Java programming language. We have used the Java Development Kit (JDK) version 11.0.11. The source code is developed in IntelliJ IDEA Community Edition 2021.1.1 and managed using Maven as the build tool.

The project's dependencies, including Hadoop libraries, are defined in the `pom.xml` file located in the root of the repository. The project is compiled and executed directly from IntelliJ IDEA.

To run the project, open the `GraphMaster` class in IntelliJ IDEA, and execute the main method. You will need to specify the following command-line arguments:

- `<input_path>` specifies the directory containing the input text files, located at `map-reduce/ probabilisticGraph /input`.
- `<output_path>` specifies the directory where the MapReduce output will be written, which will be created in the `out` folder.
- `<T>` is the minimum edge-degree threshold for vertices to be included in the results.

IntelliJ IDEA will handle the compilation and execution automatically when the main method is run. Make sure to configure the input and output paths as required for your specific run.

Note that the command-line arguments must follow the specified order and format. If any of the arguments are missing or invalid, the program will terminate with an appropriate error message.

5.2.2 Implementation. The problem at hand involves processing a probabilistic graph, where each edge connects two vertices with a probability value. The task is to compute the average degree for each vertex, considering only edges whose probability exceeds a given threshold T . The methodology proposed here utilizes a distributed MapReduce framework, implemented using Hadoop, to efficiently process and analyze large-scale graph data. The approach is divided into three main phases, each handled by separate MapReduce jobs, with intermediate results passed between the phases.

The `GraphMaster` class serves as the orchestrator of the entire MapReduce process. It manages the execution of the three phases, invoking the appropriate MapReduce jobs in sequence. The main steps executed by `GraphMaster` are:

- (1) It retrieves the command-line arguments, including the input and output directories as well as the threshold value T , which determines which edges to consider in the graph.

- (2) It initiates the first MapReduce job to compute the degree of each vertex in the graph. The input consists of edge data, and the output is a summation of edge probabilities for each vertex.
- (3) After the first job completes, the GraphMaster starts the second MapReduce job to calculate the mean degree of the graph, using the results from the first job.
- (4) Once the mean degree is computed, GraphMaster starts the third MapReduce job, which filters out the vertices with degrees lower than the mean degree.
- (5) It cleans up intermediate directories after each phase and moves the final output to the desired location.

GraphMaster coordinates these steps by configuring and executing the MapReduce jobs, ensuring that each phase depends on the results of the previous one.

The entire process consists of three MapReduce jobs, executed sequentially:

- (1) **Phase 1:** The first job computes the degree of each vertex by summing the probabilities of the edges connected to it. It produces intermediate results for each vertex.
- (2) **Phase 2:** The second job calculates the mean degree by summing the degrees from the first phase and dividing by the total number of vertices.
- (3) **Phase 3:** The third job filters out the vertices with a degree lower than the mean degree and produces the final filtered result.

After each job completes, intermediate data is written to disk and passed as input to the subsequent job. The final output consists of the vertices whose degree exceeds or equals the mean degree.

Phase 1: Calculating Vertex Degree The first MapReduce job is responsible for calculating the degree of each vertex in the graph, where the degree is defined as the sum of the probabilities of the edges connected to that vertex. The degree of each vertex is computed by the GraphMapper and GraphReducer classes.

The GraphMapper class reads each edge from the input data, which consists of lines containing two vertices and the associated probability value. It splits each line into two components (the two vertices), and for each vertex, it emits the corresponding probability value as the output. The mapper also checks if the probability value exceeds the threshold T , which is provided as a configuration parameter. If the probability is greater than or equal to T , it emits the vertex along with the corresponding probability.

The GraphReducer class receives the vertex and associated probability values emitted by the mapper. For each vertex, it sums up all the probabilities of the edges connected to that vertex, and emits the pair as output.

Phase 2: Calculating the Mean Degree The second phase of the approach calculates the mean degree of all vertices. This is done by summing the degree values from the previous phase and dividing by the total number of vertices.

The MeanMapper class receives the degree values from the output of the first MapReduce job. It emits two key-value pairs for each input line: a count of the number of vertices and the sum of the degree values. These are emitted with fixed keys ("count" and "sum") so that they can be aggregated in the reducer.

The MeanReducer class processes the count and sum values emitted by the mapper. It calculates the total sum and count of vertices and then writes these values as output. The mean degree is calculated by dividing the sum by the count, and the result is stored for use in the next phase.

Phase 3: Filtering Vertices by Mean Degree In the third phase, the vertices whose degree is greater than or equal to the mean degree calculated in Phase 2 are selected. This phase uses the results from Phase 1 and Phase 2, applying the filtering criteria to retain only those vertices with a degree greater than or equal to the mean.

The FilterMapper class takes the degree values emitted by the first reducer and writes them as key-value pairs. Each key is a vertex, and the value is the degree of that vertex. The mapper does not perform any filtering; instead, it simply passes the values along to the reducer.

The `FilterReducer` class filters out the vertices whose degree is less than the mean degree. It retrieves the mean degree from the configuration, and for each vertex, it compares the degree value to the mean. If the degree is greater than or equal to the mean, it emits the vertex along with its degree. This final output contains only the vertices that satisfy the filtering criteria.

5.2.3 Evaluation. The experiments were executed using the dataset ‘collins.txt’ with a parameter setting of $T = 0.8$.

To provide a comprehensive overview, the results of the experiment, including the detailed values derived from the dataset, are presented in subsection C.1 in a 2-column format.

6 Conclusion

TODO: In this document we have presented our solutions and rational for solving the first assignment of the M.Sc. course on *Technologies for Big Data Analysis*, offered by the *DWS M.Sc. Program*.

A Problem 1

In the following 8-columned pages, the experimental results of section’s 2 problem are presented, as the raw output of the MapReduce execution.

a10c: 34	a12g: 47	a3y: 1193	a6k: 151	a8h: 328	b10r: 33	b3w: 35	b5s: 1288
a10d: 486	a12l: 20	a4a: 458	a6l: 192	a8l: 125	b10s: 227	b3x: 66	b5t: 212
a10e: 386	a12n: 65	a4d: 166	a6m: 20	a8m: 239	b10t: 43	b3y: 223	b5w: 111
a10g: 175	a12s: 139	a4e: 249	a6n: 456	a8n: 982	b10y: 10	b4d: 268	b5y: 101
a10h: 58	a12t: 59	a4h: 13	a6p: 84	a8r: 98	b11a: 10	b4e: 883	b6a: 42
a10l: 131	a12y: 36	a4n: 15	a6r: 523	a8s: 469	b11d: 79	b4f: 10	b6d: 865
a10m: 15	a13e: 10	a4o: 773	a6s: 1197	a8t: 440	b11g: 15	b4g: 96	b6e: 3041
a10n: 260	a13g: 19	a4s: 483	a6t: 811	a8x: 10	b11m: 34	b4h: 538	b6f: 52
a10p: 48	a13n: 22	a4t: 47	a6w: 1078	a8y: 328	b11n: 21	b4k: 969	b6g: 84
a10r: 68	a13s: 112	a4y: 1588	a6y: 204	a9a: 27	b11o: 48	b4l: 288	b6h: 123
a10s: 429	a13t: 36	a5a: 41	a7a: 431	a9c: 23	b11s: 75	b4m: 16	b6i: 20
a10t: 72	a13y: 36	a5d: 1360	a7c: 52	a9d: 531	b11t: 13	b4n: 2688	b6l: 39
a10y: 190	a14c: 12	a5e: 1377	a7d: 1076	a9e: 369	b11y: 19	b4r: 102	b6m: 41
a10z: 50	a14e: 16	a5g: 841	a7e: 685	a9g: 548	b12d: 36	b4s: 119	b6n: 329
a11a: 36	a14n: 114	a5i: 11	a7g: 71	a9l: 62	b12e: 20	b4t: 458	b6r: 436
a11c: 36	a14s: 16	a5l: 112	a7h: 12	a9n: 555	b12g: 10	b4w: 98	b6s: 390
a11d: 165	a14t: 17	a5m: 56	a7l: 116	a9r: 235	b12l: 18	b4y: 459	b6t: 347
a11e: 136	a14y: 38	a5n: 916	a7n: 83	a9s: 622	b12s: 25	b5a: 77	b6u: 17
a11g: 101	a15s: 33	a5r: 1604	a7r: 843	a9t: 287	b12t: 10	b5d: 934	b6w: 15
a11n: 202	a16s: 18	a5s: 213	a7s: 1203	a9v: 38	b12w: 10	b5e: 330	b6y: 181
a11p: 16	a3d: 38224	a5t: 1767	a7t: 1011	a9y: 286	b12y: 21	b5f: 29	b7a: 31
a11r: 13	a3e: 3767	a5w: 94	a7x: 17	b10a: 17	b13e: 10	b5g: 1080	b7c: 25
a11s: 266	a3k: 248	a5y: 201	a7y: 756	b10d: 86	b13v: 13	b5h: 103	b7d: 248
a11t: 115	a3l: 4077	a6a: 112	a8a: 28	b10e: 58	b3d: 352	b5k: 342	b7e: 1005
a11y: 119	a3m: 353	a6d: 785	a8c: 37	b10g: 28	b3e: 11	b5l: 11	b7g: 339
a12c: 18	a3o: 106	a6e: 615	a8d: 1790	b10l: 20	b3g: 145	b5m: 29	b7h: 371
a12d: 94	a3r: 198	a6g: 214	a8e: 386	b10m: 12	b3r: 22	b5n: 1190	b7i: 31
a12e: 95	a3t: 428	a6h: 21	a8g: 625	b10n: 13	b3t: 5674	b5r: 60	b7k: 28

b7l: 16	c11d: 116	c4b: 72	c7c: 148	d10l: 29	d3s: 11	d7e: 1011	e10t: 195
b7m: 61	c11e: 272	c4d: 333	c7d: 1329	d10n: 175	d3y: 882	d7g: 439	e10y: 610
b7n: 861	c11g: 141	c4e: 2548	c7e: 561	d10r: 15	d4a: 19	d7l: 18	e11a: 49
b7o: 15	c11l: 92	c4f: 18	c7g: 328	d10s: 211	d4d: 284	d7n: 43	e11d: 331
b7r: 238	c11n: 403	c4k: 36	c7h: 15	d10t: 120	d4e: 602	d7o: 10	e11e: 20
b7s: 741	c11p: 12	c4l: 246	c7k: 80	d10v: 44	d4g: 27	d7p: 56	e11g: 29
b7t: 494	c11r: 16	c4m: 88	c7l: 690	d10y: 259	d4h: 19	d7r: 107	e11l: 62
b7u: 16	c11s: 341	c4n: 92	c7m: 35	d11a: 13	d4k: 187	d7s: 665	e11m: 20
b7y: 196	c11t: 74	c4p: 105	c7n: 703	d11d: 205	d4l: 170	d7t: 220	e11n: 195
b8a: 121	c11y: 181	c4r: 31	c7o: 27	d11e: 106	d4m: 29	d7v: 366	e11s: 134
b8d: 308	c12d: 107	c4s: 42	c7r: 658	d11g: 78	d4n: 1183	d7y: 183	e11t: 95
b8e: 83	c12e: 284	c4t: 739	c7s: 834	d11h: 32	d4p: 258	d8a: 46	e11y: 115
b8g: 451	c12g: 46	c4w: 12	c7t: 612	d11l: 23	d4r: 935	d8c: 93	e12a: 11
b8h: 15	c12h: 11	c4y: 285	c7x: 23	d11m: 17	d4s: 900	d8d: 903	e12c: 16
b8l: 55	c12l: 15	c5a: 45	c7y: 1192	d11n: 352	d4t: 713	d8e: 536	e12d: 49
b8m: 41	c12n: 673	c5c: 10	c8a: 125	d11s: 176	d4u: 19	d8g: 305	e12g: 79
b8n: 53	c12r: 10	c5d: 2428	c8c: 46	d11t: 121	d4w: 203	d8h: 52	e12l: 13
b8o: 105	c12s: 173	c5e: 804	c8d: 582	d11y: 63	d4y: 161	d8k: 15	e12n: 38
b8r: 55	c12t: 45	c5f: 320	c8e: 638	d12d: 112	d5a: 11	d8l: 119	e12s: 88
b8s: 795	c12y: 124	c5g: 24	c8g: 570	d12e: 70	d5d: 114	d8m: 37	e12t: 23
b8t: 62	c13d: 80	c5h: 168	c8h: 26	d12g: 44	d5e: 296	d8n: 272	e12y: 20
b8v: 148	c13e: 35	c5i: 18	c8l: 573	d12l: 21	d5g: 250	d8r: 296	e13n: 15
b8y: 78	c13g: 61	c5k: 237	c8n: 523	d12n: 162	d5h: 374	d8s: 475	e13s: 59
b9a: 26	c13l: 29	c5l: 318	c8o: 18	d12s: 168	d5k: 109	d8t: 183	e13t: 75
b9d: 95	c13n: 220	c5m: 77	c8r: 285	d12t: 32	d5l: 77	d8v: 284	e13y: 91
b9e: 232	c13s: 379	c5n: 161	c8s: 2081	d12y: 37	d5m: 43	d8y: 138	e14g: 10
b9g: 271	c13y: 83	c5p: 33	c8t: 341	d13d: 84	d5n: 209	d9a: 31	e14s: 24
b9h: 18	c14a: 14	c5r: 479	c8x: 16	d13e: 52	d5s: 415	d9d: 793	e14y: 14
b9i: 161	c14c: 94	c5s: 1360	c8y: 394	d13g: 13	d5t: 305	d9e: 326	e3a: 11
b9l: 175	c14e: 47	c5t: 1031	c9a: 39	d13n: 63	d5y: 245	d9g: 169	e3c: 18
b9n: 274	c14l: 93	c5y: 130	c9d: 1289	d13s: 29	d6a: 39	d9m: 14	e3d: 456
b9r: 19	c14n: 20	c6a: 44	c9e: 440	d13t: 13	d6d: 313	d9n: 202	e3e: 137
b9s: 163	c14s: 84	c6c: 40	c9g: 137	d14d: 17	d6e: 549	d9r: 48	e3r: 45
b9t: 150	c14y: 29	c6d: 871	c9l: 41	d14g: 16	d6g: 566	d9s: 855	e3t: 67
b9y: 19	c15d: 11	c6e: 1144	c9m: 42	d14n: 27	d6h: 10	d9t: 470	e4a: 12
c10a: 49	c15e: 12	c6g: 247	c9n: 616	d14s: 10	d6i: 14	d9v: 26	e4e: 304
c10d: 460	c15s: 24	c6h: 125	c9r: 485	d14t: 11	d6l: 146	d9y: 257	e4h: 409
c10e: 395	c15y: 18	c6l: 61	c9s: 836	d14y: 13	d6m: 27	e10c: 57	e4l: 53
c10g: 362	c16e: 39	c6m: 30	c9t: 198	d15n: 31	d6n: 177	e10d: 104	e4n: 941
c10l: 129	c16f: 10	c6n: 676	c9v: 10	d15s: 19	d6r: 617	e10e: 256	e4r: 253
c10n: 689	c3b: 34	c6r: 706	c9y: 381	d3c: 10	d6s: 281	e10g: 553	e4s: 1113
c10r: 18	c3n: 872	c6s: 929	d10a: 113	d3d: 1874	d6t: 341	e10h: 23	e4t: 149
c10s: 818	c3p: 117	c6t: 568	d10c: 142	d3e: 335	d6y: 154	e10l: 67	e4y: 129
c10t: 97	c3t: 210	c6x: 23	d10d: 345	d3g: 71	d7b: 13	e10m: 91	e5a: 13
c10y: 256	c3y: 140	c6y: 363	d10e: 169	d3m: 55	d7c: 21	e10n: 431	e5d: 43
c11a: 21	c4a: 35	c7a: 23	d10g: 194	d3n: 29	d7d: 989	e10s: 136	e5e: 56

e5h: 117	e8t: 40	f4d: 493	f7m: 220	g11s: 71	g6r: 47	h11a: 19	h5r: 193
e5k: 85	e8y: 174	f4e: 2351	f7n: 410	g12d: 10	g6s: 717	h11c: 13	h5s: 1072
e5l: 164	e9c: 42	f4g: 31	f7r: 333	g12l: 16	g6t: 10	h11d: 17	h5t: 311
e5n: 46	e9d: 544	f4h: 16	f7s: 1066	g12s: 74	g6y: 229	h11e: 143	h5y: 509
e5r: 210	e9e: 386	f4i: 27	f7t: 43	g13y: 10	g7a: 83	h11g: 21	h6a: 30
e5s: 98	e9g: 244	f4k: 35	f7y: 265	g14g: 10	g7d: 366	h11n: 16	h6c: 33
e5t: 327	e9h: 75	f4l: 859	f8d: 604	g18g: 13	g7e: 126	h11s: 32	h6d: 321
e5w: 87	e9l: 122	f4m: 6376	f8e: 190	g3d: 305	g7g: 314	h11y: 13	h6e: 212
e5y: 1240	e9n: 283	f4p: 28	f8g: 218	g3m: 13	g7h: 19	h12a: 17	h6g: 718
e6a: 13	e9r: 31	f4r: 444	f8h: 39	g3n: 54	g7l: 848	h12c: 12	h6h: 170
e6d: 174	e9s: 330	f4s: 73	f8k: 11	g3p: 44	g7m: 25	h12f: 55	h6i: 14
e6e: 651	e9t: 153	f4t: 1443	f8l: 56	g3s: 11	g7n: 29	h12g: 12	h6n: 214
e6g: 39	e9y: 634	f4w: 96	f8m: 21	g3t: 752	g7p: 15	h12s: 90	h6r: 443
e6h: 189	f10d: 228	f4x: 11	f8n: 226	g3y: 33	g7r: 167	h13d: 13	h6s: 1051
e6i: 12	f10e: 83	f4y: 21	f8r: 152	g4d: 855	g7s: 328	h13s: 20	h6t: 96
e6m: 10	f10g: 95	f5a: 22	f8s: 473	g4e: 1349	g7t: 31	h3d: 7393	h6w: 55
e6n: 28	f10h: 29	f5d: 1113	f8t: 118	g4f: 13	g7y: 216	h3e: 11	h6y: 274
e6r: 335	f10n: 128	f5e: 402	f8y: 223	g4l: 185	g8c: 29	h3m: 5193	h7a: 16
e6s: 496	f10p: 42	f5g: 32	f9a: 17	g4m: 20	g8d: 221	h3p: 37	h7d: 544
e6t: 710	f10r: 20	f5h: 404	f9c: 12	g4n: 91	g8e: 185	h3r: 5252	h7e: 44
e6y: 213	f10s: 124	f5k: 140	f9d: 147	g4p: 14	g8g: 232	h3s: 11727	h7f: 1487
e7a: 18	f10t: 63	f5l: 132	f9e: 131	g4s: 269	g8l: 37	h3t: 268	h7g: 535
e7c: 69	f10v: 12	f5n: 23	f9g: 266	g4t: 44	g8n: 73	h3w: 1296	h7h: 13
e7d: 1198	f10y: 242	f5r: 353	f9h: 26	g4w: 255	g8r: 139	h3y: 47	h7l: 48
e7e: 588	f11d: 19	f5s: 867	f9k: 31	g4y: 110	g8s: 306	h4d: 2139	h7m: 25
e7g: 279	f11e: 46	f5t: 1749	f9l: 51	g5a: 40	g8t: 83	h4e: 4797	h7n: 48
e7h: 213	f11g: 73	f5y: 346	f9n: 454	g5d: 278	g8y: 37	h4f: 251	h7r: 466
e7l: 29	f11l: 21	f6a: 61	f9o: 16	g5e: 276	g9a: 19	h4g: 59	h7s: 490
e7m: 11	f11n: 42	f6d: 958	f9r: 24	g5f: 47	g9d: 33	h4h: 286	h7t: 118
e7n: 255	f11s: 142	f6e: 500	f9s: 309	g5g: 369	g9e: 65	h4k: 11	h7y: 683
e7o: 10	f11t: 15	f6g: 183	f9t: 53	g5h: 15	g9g: 188	h4l: 277	h8c: 47
e7r: 569	f11y: 59	f6h: 1159	f9y: 59	g5i: 10	g9n: 179	h4m: 38	h8d: 313
e7s: 575	f12g: 14	f6l: 60	g10a: 20	g5m: 59	g9r: 45	h4n: 22	h8e: 182
e7t: 209	f12n: 11	f6n: 122	g10d: 16	g5n: 594	g9s: 160	h4o: 53	h8g: 47
e7v: 28	f12s: 33	f6o: 13	g10g: 52	g5p: 168	g9t: 37	h4p: 259	h8h: 10
e7y: 234	f13s: 10	f6r: 892	g10n: 37	g5s: 549	g9y: 190	h4r: 557	h8l: 66
e8a: 11	f3d: 18	f6s: 520	g10r: 12	g5t: 935	h10a: 55	h4s: 66	h8n: 150
e8c: 152	f3e: 47	f6t: 319	g10s: 120	g5y: 136	h10c: 17	h4t: 229	h8o: 22
e8d: 864	f3g: 340	f6w: 344	g10t: 643	g6a: 13	h10d: 35	h4y: 50	h8s: 254
e8e: 572	f3n: 26	f6y: 435	g10y: 40	g6d: 281	h10e: 22	h5d: 709	h8t: 33
e8g: 232	f3o: 17	f7a: 76	g11d: 60	g6e: 336	h10g: 26	h5e: 1028	h8y: 89
e8l: 107	f3r: 7246	f7d: 590	g11g: 22	g6g: 256	h10h: 21	h5h: 23	h9a: 19
e8m: 19	f3t: 128	f7e: 324	g11l: 21	g6h: 223	h10l: 76	h5i: 10	h9c: 10
e8n: 348	f3w: 453	f7g: 756	g11m: 126	g6l: 22	h10n: 44	h5l: 22	h9d: 139
e8r: 62	f3x: 44	f7h: 48	g11n: 89	g6n: 305	h10s: 207	h5n: 184	h9e: 136
e8s: 531	f3y: 33	f7l: 282	g11r: 20	g6p: 63	h10y: 38	h5o: 10	h9g: 70

h9k: 10	i3k: 12	i8l: 221	j7e: 83	k7h: 15	l4e: 3014	l8g: 201	m13s: 160
h9n: 61	i3l: 278	i8m: 41	j7g: 71	k7m: 24	l4g: 942	l8n: 79	m14s: 65
h9s: 256	i3n: 24	i8n: 96	j7n: 87	k7n: 117	l4h: 10	l8r: 97	m15s: 10
h9t: 18	i3s: 2045	i8r: 90	j7r: 20	k7s: 39	l4k: 658	l8s: 142	m3b: 26
h9y: 121	i4a: 188	i8s: 247	j7s: 26	k7v: 448	l4n: 42	l8t: 42	m3d: 61
i10d: 309	i4e: 25	i8t: 414	j7y: 127	k8a: 21	l4p: 53	l8y: 75	m3n: 2654
i10e: 645	i4h: 11	i8y: 305	j8e: 15	k8f: 11	l4s: 1182	l9a: 102	m3p: 40
i10g: 134	i4n: 128	i9a: 14	j8g: 12	k8g: 26	l4t: 1734	l9c: 23	m3s: 58
i10l: 234	i4o: 2120	i9d: 418	j8l: 26	k8s: 131	l4y: 164	l9d: 105	m3t: 547
i10n: 306	i4s: 12	i9e: 538	j8n: 52	k8v: 70	l5a: 17	l9e: 45	m3y: 2535
i10s: 253	i4t: 50	i9g: 250	j8s: 46	k8y: 70	l5c: 15	l9g: 121	m4d: 514
i10t: 99	i5a: 23	i9h: 35	j8t: 34	k9e: 71	l5d: 371	l9n: 29	m4e: 3829
i10y: 163	i5c: 25	i9l: 24	j8y: 72	k9i: 17	l5e: 965	l9r: 46	m4h: 671
i11d: 212	i5e: 118	i9n: 557	j9d: 21	k9s: 32	l5g: 114	l9s: 137	m4k: 98
i11e: 172	i5h: 20	i9r: 89	j9n: 108	k9t: 22	l5h: 291	l9y: 53	m4l: 32
i11g: 127	i5l: 32	i9s: 381	j9s: 24	l10d: 24	l5l: 303	m10c: 11	m4n: 233
i11l: 15	i5n: 39	i9t: 371	j9y: 17	l10e: 52	l5n: 102	m10d: 82	m4s: 396
i11m: 10	i5r: 65	i9w: 34	k11a: 25	l10g: 13	l5r: 934	m10e: 72	m4t: 2077
i11n: 300	i5s: 91	i9y: 124	k11d: 10	l10n: 42	l5s: 739	m10g: 27	m4y: 1246
i11r: 23	i5t: 20	j10n: 23	k11n: 15	l10p: 56	l5t: 480	m10l: 50	m5a: 308
i11s: 229	i5x: 17	j10s: 33	k3y: 23	l10s: 135	l5y: 49	m10m: 13	m5c: 59
i11t: 191	i5y: 73	j12n: 27	k4d: 194	l10t: 27	l6a: 41	m10n: 58	m5d: 312
i11y: 354	i6a: 11	j13e: 10	k4e: 135	l10y: 15	l6d: 1083	m10r: 12	m5e: 66
i12d: 19	i6c: 10	j13n: 15	k4g: 211	l11d: 16	l6e: 1215	m10s: 199	m5h: 389
i12e: 359	i6d: 291	j3w: 33	k4l: 64	l11e: 112	l6g: 269	m10t: 63	m5l: 109
i12g: 33	i6e: 216	j3y: 103	k4n: 35	l11g: 15	l6h: 66	m10y: 69	m5m: 27
i12l: 55	i6f: 264	j4e: 80	k4p: 151	l11n: 56	l6l: 31	m11a: 133	m5n: 51
i12n: 505	i6m: 34	j4k: 10	k4s: 59	l11s: 40	l6n: 315	m11d: 16	m5o: 12
i12s: 141	i6n: 64	j4n: 208	k4t: 279	l11y: 10	l6r: 930	m11e: 51	m5r: 150
i12t: 52	i6s: 110	j4p: 13	k4v: 17	l12a: 19	l6s: 314	m11g: 26	m5s: 701
i12y: 145	i6t: 119	j4s: 45	k4w: 1529	l12d: 13	l6t: 52	m11i: 81	m5t: 832
i13e: 100	i6y: 127	j4t: 773	k5e: 45	l12g: 11	l6x: 25	m11n: 10	m5y: 567
i13l: 51	i7a: 32	j4y: 62	k5k: 12	l12n: 22	l6y: 241	m11r: 10	m6a: 46
i13n: 35	i7d: 420	j5e: 106	k5l: 48	l12s: 129	l7c: 10	m11s: 90	m6d: 390
i13r: 20	i7e: 352	j5n: 22	k5n: 379	l13n: 11	l7d: 391	m11t: 20	m6e: 621
i13s: 32	i7g: 18	j5s: 137	k5s: 255	l13s: 19	l7e: 80	m11y: 26	m6f: 223
i13t: 33	i7h: 14	j5t: 329	k5y: 18	l3d: 274	l7g: 952	m12d: 10	m6g: 365
i13y: 222	i7l: 35	j5y: 14	k6a: 26	l3e: 109	l7l: 90	m12e: 155	m6l: 117
i14e: 26	i7n: 37	j6d: 170	k6d: 279	l3g: 157	l7n: 92	m12h: 22	m6m: 62
i14n: 20	i7s: 296	j6e: 10	k6r: 22	l3p: 63	l7r: 47	m12n: 24	m6n: 227
i14s: 16	i7t: 134	j6g: 19	k6s: 52	l3s: 20	l7s: 505	m12r: 32	m6o: 72
i14y: 13	i7y: 44	j6h: 47	k6y: 118	l3t: 573	l7t: 46	m12s: 88	m6r: 1005
i15y: 17	i8a: 25	j6l: 53	k7a: 20	l3w: 486	l7v: 10	m12y: 31	m6s: 452
i16e: 39	i8d: 845	j6s: 239	k7d: 49	l3y: 289	l7y: 211	m13g: 38	m6t: 569
i3e: 78	i8e: 317	j6t: 39	k7e: 29	l4b: 213	l8d: 221	m13l: 11	m6u: 10
i3i: 91	i8g: 88	j6y: 49	k7g: 213	l4d: 791	l8e: 195	m13n: 29	m6v: 12

m6w: 803	n10c: 16	n6d: 138	o10h: 28	o6a: 76	o9w: 19	p3n: 31	p6y: 342
m6x: 10	n10d: 20	n6e: 345	o10l: 26	o6d: 243	o9y: 56	p3r: 88	p7a: 93
m6y: 443	n10e: 41	n6g: 13	o10n: 177	o6e: 181	p10a: 38	p3s: 170	p7c: 94
m7a: 163	n10g: 23	n6i: 10	o10s: 177	o6g: 17	p10c: 17	p3t: 465	p7d: 592
m7c: 23	n10h: 21	n6l: 107	o10y: 31	o6k: 100	p10d: 138	p3y: 123	p7e: 843
m7d: 388	n10k: 18	n6n: 210	o11d: 36	o6n: 171	p10e: 184	p4a: 67	p7g: 540
m7e: 302	n10n: 29	n6r: 379	o11g: 54	o6r: 13	p10g: 464	p4d: 122	p7k: 18
m7g: 624	n10s: 57	n6s: 243	o11n: 51	o6s: 913	p10h: 54	p4e: 410	p7l: 134
m7h: 27	n10y: 30	n6t: 10	o11s: 94	o6t: 159	p10l: 128	p4h: 115	p7m: 147
m7l: 178	n11g: 24	n6w: 77	o11y: 79	o6y: 47	p10m: 121	p4k: 99	p7n: 223
m7m: 40	n11m: 22	n6y: 268	o12a: 14	o6z: 22	p10n: 521	p4l: 60	p7r: 246
m7n: 265	n11n: 11	n7a: 1119	o12d: 30	o7a: 31	p10r: 124	p4m: 39	p7s: 1374
m7r: 38	n11s: 35	n7d: 203	o12g: 17	o7c: 14	p10s: 413	p4n: 463	p7t: 1435
m7s: 1188	n11t: 16	n7e: 76	o12n: 94	o7d: 336	p10t: 142	p4p: 11	p7x: 39
m7t: 11	n11y: 62	n7g: 662	o12s: 23	o7e: 273	p10y: 317	p4r: 186	p7y: 237
m7y: 221	n12d: 18	n7k: 12	o12y: 91	o7f: 26	p11c: 22	p4s: 205	p8a: 149
m8a: 24	n12g: 18	n7l: 164	o13s: 108	o7g: 151	p11d: 50	p4t: 1040	p8c: 136
m8c: 41	n12s: 83	n7n: 10	o3d: 1140	o7k: 10	p11e: 166	p4y: 256	p8d: 819
m8d: 222	n13d: 18	n7r: 179	o3e: 3233	o7n: 232	p11g: 88	p5a: 255	p8e: 1405
m8e: 505	n13n: 33	n7s: 364	o3f: 625	o7r: 491	p11n: 117	p5c: 17	p8g: 312
m8g: 123	n13s: 12	n7t: 73	o3g: 15	o7s: 236	p11r: 25	p5d: 98	p8h: 16
m8i: 56	n15d: 12	n7y: 71	o3h: 13	o7t: 11	p11s: 289	p5e: 1330	p8i: 16
m8k: 29	n17s: 13	n8a: 25	o3k: 27	o7y: 47	p11t: 24	p5f: 30	p8k: 12
m8l: 137	n3p: 10	n8d: 33	o3l: 33	o8a: 30	p11y: 240	p5h: 143	p8l: 356
m8m: 17	n3r: 275	n8e: 134	o3n: 777	o8d: 564	p12a: 159	p5k: 10	p8m: 66
m8n: 158	n3t: 6590	n8g: 64	o3r: 1064	o8e: 194	p12d: 37	p5l: 69	p8n: 539
m8r: 283	n3w: 2862	n8l: 288	o3t: 1956	o8g: 69	p12e: 46	p5n: 102	p8r: 160
m8s: 535	n4d: 156	n8n: 619	o4d: 13	o8h: 12	p12g: 14	p5o: 26	p8s: 2343
m8t: 448	n4e: 683	n8p: 11	o4e: 567	o8k: 37	p12l: 106	p5p: 31	p8t: 221
m8y: 608	n4k: 193	n8r: 84	o4h: 27	o8l: 189	p12n: 115	p5r: 766	p8y: 390
m9a: 25	n4l: 58	n8s: 932	o4l: 24	o8m: 34	p12r: 13	p5s: 870	p9a: 56
m9c: 18	n4r: 286	n8y: 47	o4n: 302	o8n: 59	p12s: 131	p5t: 299	p9c: 35
m9d: 138	n4s: 237	n9a: 35	o4o: 159	o8r: 29	p12t: 18	p5y: 332	p9d: 714
m9e: 125	n4t: 281	n9d: 68	o4r: 1263	o8s: 467	p12y: 223	p6a: 44	p9e: 216
m9g: 46	n4y: 35	n9e: 56	o4s: 213	o8t: 117	p13e: 19	p6c: 307	p9g: 257
m9h: 10	n5d: 75	n9g: 16	o4y: 1890	o8w: 18	p13g: 10	p6d: 1413	p9h: 43
m9i: 28	n5e: 457	n9i: 66	o5a: 16	o8y: 100	p13k: 11	p6e: 4994	p9l: 405
m9l: 28	n5h: 222	n9l: 17	o5d: 19	o9a: 13	p13l: 20	p6g: 85	p9m: 20
m9m: 24	n5i: 10	n9n: 23	o5e: 27	o9d: 111	p13n: 38	p6h: 106	p9n: 169
m9n: 49	n5l: 79	n9r: 23	o5g: 44	o9e: 220	p13r: 22	p6l: 75	p9o: 13
m9o: 30	n5o: 22	n9s: 168	o5m: 23	o9g: 218	p13s: 62	p6n: 277	p9r: 73
m9r: 36	n5r: 590	n9t: 53	o5n: 465	o9h: 11	p13y: 20	p6o: 11	p9s: 633
m9s: 505	n5s: 189	n9y: 486	o5r: 2249	o9n: 252	p14s: 14	p6r: 213	p9t: 494
m9t: 124	n5t: 335	o10d: 65	o5s: 20	o9p: 28	p15s: 11	p6s: 813	p9y: 262
m9y: 92	n5y: 58	o10e: 91	o5t: 173	o9s: 367	p3a: 11	p6t: 205	q10d: 23
n10a: 11	n6a: 20	o10g: 26	o5y: 12	o9t: 11	p3d: 19	p6w: 26	q10g: 13

q10s: 20	r12n: 115	r5t: 714	r9n: 141	s13r: 10	s5f: 189	s7w: 29	t11e: 135
q11g: 27	r12s: 72	r5w: 12	r9r: 21	s13s: 67	s5g: 42	s7y: 655	t11g: 36
q13n: 10	r12y: 59	r5y: 426	r9s: 475	s13t: 43	s5h: 371	s8a: 29	t11h: 33
q13r: 16	r13d: 11	r6a: 185	r9t: 164	s13y: 51	s5k: 522	s8c: 43	t11l: 27
q14s: 26	r13e: 17	r6d: 791	r9y: 15	s14e: 17	s5l: 2431	s8d: 882	t11n: 68
q3e: 16	r13n: 13	r6e: 290	s10c: 158	s14n: 11	s5m: 131	s8e: 550	t11r: 11
q4t: 11	r13s: 25	r6f: 65	s10d: 321	s14s: 29	s5n: 545	s8g: 1342	t11s: 324
q5e: 500	r13y: 71	r6g: 174	s10e: 125	s14t: 55	s5o: 17	s8h: 211	t11t: 32
q5k: 75	r14e: 39	r6k: 46	s10g: 230	s14y: 47	s5p: 350	s8k: 203	t11y: 34
q5n: 18	r14g: 10	r6l: 115	s10l: 69	s15n: 10	s5r: 145	s8l: 219	t12d: 29
q5r: 13	r14n: 87	r6m: 135	s10m: 50	s15r: 10	s5s: 1274	s8m: 19	t12e: 58
q5t: 145	r14s: 24	r6n: 775	s10n: 196	s15y: 10	s5t: 1035	s8n: 509	t12g: 18
q6s: 12	r14y: 39	r6r: 415	s10o: 32	s16n: 14	s5w: 24	s8r: 325	t12h: 23
q6y: 15	r15s: 104	r6s: 464	s10r: 44	s16s: 10	s5y: 451	s8s: 1622	t12n: 10
q7l: 31	r3b: 10	r6t: 710	s10s: 469	s3a: 74	s6a: 44	s8t: 287	t12s: 121
q7r: 62	r3d: 252	r6v: 715	s10t: 204	s3b: 18	s6b: 12	s8v: 15	t12y: 42
q7y: 276	r3e: 18	r6w: 43	s10y: 114	s3c: 72	s6c: 51	s8x: 30	t13d: 10
q8d: 17	r3g: 12	r6y: 402	s11a: 24	s3d: 87	s6d: 3507	s8y: 1160	t13e: 14
q8n: 339	r3m: 24	r7a: 32	s11c: 28	s3e: 4950	s6e: 1353	s9a: 35	t13n: 16
q8s: 77	r3n: 458	r7d: 1139	s11d: 98	s3m: 55	s6g: 824	s9c: 50	t13s: 23
q8y: 38	r3p: 15	r7e: 542	s11e: 98	s3n: 409	s6h: 368	s9d: 806	t13y: 18
q9d: 28	r3t: 11	r7g: 496	s11g: 154	s3r: 176	s6k: 162	s9e: 371	t14n: 32
q9g: 18	r3w: 79	r7h: 25	s11h: 43	s3t: 801	s6l: 238	s9g: 1178	t14s: 22
q9s: 218	r4d: 472	r7i: 15	s11l: 141	s3w: 600	s6m: 373	s9h: 14	t15n: 11
r10d: 398	r4e: 1174	r7l: 179	s11m: 17	s3x: 160	s6n: 423	s9i: 67	t3a: 104
r10e: 234	r4f: 30	r7n: 502	s11n: 206	s3y: 858	s6r: 559	s9k: 21	t3e: 79438
r10g: 175	r4g: 78	r7r: 111	s11r: 21	s4a: 109	s6s: 2097	s9l: 98	t3n: 223
r10l: 75	r4h: 129	r7s: 1095	s11s: 151	s4b: 23	s6t: 927	s9m: 13	t3o: 1608
r10m: 39	r4k: 124	r7t: 329	s11t: 51	s4d: 3697	s6w: 135	s9n: 429	t3p: 45
r10n: 640	r4l: 129	r7y: 473	s11w: 12	s4e: 3848	s6y: 689	s9o: 13	t3t: 20
r10r: 11	r4m: 905	r8c: 69	s11y: 108	s4f: 26	s7a: 153	s9r: 137	t3x: 75
r10s: 168	r4n: 90	r8d: 2212	s12d: 79	s4g: 123	s7b: 12	s9s: 1044	t3y: 137
r10t: 29	r4o: 22	r8e: 325	s12e: 107	s4h: 1441	s7c: 26	s9t: 194	t4d: 590
r10y: 99	r4p: 11	r8g: 237	s12g: 40	s4k: 207	s7d: 1327	s9y: 486	t4e: 2747
r11d: 110	r4r: 49	r8h: 68	s12h: 15	s4l: 309	s7e: 1364	t10a: 13	t4k: 903
r11e: 78	r4s: 142	r8l: 24	s12i: 17	s4m: 135	s7g: 1121	t10d: 74	t4l: 849
r11g: 87	r4t: 262	r8n: 327	s12n: 81	s4n: 1527	s7h: 163	t10e: 100	t4m: 2353
r11h: 14	r4y: 39	r8r: 184	s12r: 15	s4p: 361	s7i: 13	t10g: 49	t4n: 3322
r11l: 13	r5d: 675	r8s: 560	s12s: 239	s4r: 148	s7l: 555	t10h: 15	t4p: 45
r11n: 151	r5e: 195	r8t: 254	s12t: 10	s4s: 446	s7m: 74	t10l: 32	t4r: 83
r11s: 259	r5h: 130	r8y: 147	s12y: 148	s4t: 1025	s7n: 390	t10n: 41	t4s: 4312
r11t: 28	r5l: 142	r9c: 15	s13d: 49	s4w: 355	s7o: 14	t10r: 53	t4t: 12263
r11y: 55	r5m: 20	r9d: 317	s13e: 38	s4y: 85	s7r: 627	t10s: 521	t4u: 40
r12d: 15	r5n: 75	r9e: 186	s13g: 17	s5a: 449	s7s: 1252	t10t: 128	t4y: 3875
r12e: 26	r5r: 210	r9g: 642	s13l: 15	s5d: 886	s7t: 434	t10y: 60	t5a: 66
r12g: 29	r5s: 525	r9m: 32	s13n: 29	s5e: 2969	s7v: 34	t11d: 112	t5b: 50

t5c: 30	t8d: 414	u12y: 82	u7y: 591	v5r: 25	v9s: 164	w5t: 165	w9t: 28
t5d: 552	t8e: 370	u13d: 21	u8d: 65	v5s: 296	v9y: 46	w5y: 84	w9y: 21
t5e: 6460	t8g: 350	u13e: 32	u8e: 44	v5t: 134	w10d: 28	w6d: 743	x3i: 74
t5f: 10	t8h: 15	u13g: 85	u8g: 38	v5x: 31	w10e: 13	w6e: 45	x3v: 34
t5g: 303	t8l: 76	u13s: 10	u8l: 16	v6a: 77	w10g: 28	w6g: 66	x3x: 19
t5h: 365	t8n: 157	u13y: 43	u8n: 50	v6d: 93	w10l: 18	w6h: 89	x4i: 56
t5k: 820	t8o: 31	u14e: 21	u8s: 67	v6e: 97	w10n: 180	w6m: 25	x4y: 30
t5l: 88	t8r: 292	u14y: 16	u8y: 68	v6i: 214	w10r: 21	w6n: 580	x5i: 33
t5n: 516	t8s: 365	u15y: 10	u9d: 228	v6l: 144	w10s: 62	w6r: 152	x5s: 53
t5p: 12	t8t: 34	u16l: 15	u9e: 42	v6m: 17	w11d: 34	w6s: 495	y10s: 20
t5r: 3002	t8w: 120	u3e: 313	u9g: 41	v6n: 29	w11g: 21	w6t: 114	y3s: 675
t5s: 1525	t8y: 235	u3m: 11	u9l: 77	v6r: 17	w11s: 45	w6w: 187	y3t: 476
t5t: 146	t9c: 48	u4a: 12	u9n: 40	v6s: 372	w12d: 11	w6y: 187	y3u: 5361
t5w: 142	t9d: 151	u4d: 271	u9s: 56	v6t: 33	w3b: 20	w7d: 411	y4d: 94
t5y: 165	t9e: 291	u4h: 18	u9t: 10	v6y: 118	w3n: 74	w7e: 173	y4k: 193
t6a: 40	t9g: 139	u4n: 1106	u9y: 56	v7a: 37	w3o: 2992	w7g: 745	y4r: 1539
t6d: 1075	t9h: 45	u4o: 13	v10d: 18	v7d: 31	w3r: 844	w7l: 10	y5d: 33
t6e: 817	t9k: 31	u5d: 52	v10e: 18	v7e: 312	w3s: 11367	w7m: 60	y5e: 119
t6f: 115	t9l: 16	u5e: 192	v10n: 14	v7g: 42	w3t: 64	w7n: 260	y5g: 622
t6g: 502	t9n: 39	u5g: 47	v10r: 16	v7k: 10	w3x: 21	w7r: 503	y5h: 65
t6h: 658	t9o: 11	u5l: 497	v10s: 72	v7n: 31	w3y: 1522	w7s: 213	y5l: 76
t6l: 94	t9r: 13	u5n: 265	v10y: 46	v7o: 10	w4d: 469	w7t: 1070	y5s: 656
t6n: 384	t9s: 250	u5r: 1283	v11n: 12	v7r: 72	w4e: 4923	w7w: 34	y5v: 11
t6r: 481	t9t: 406	u5s: 17	v11y: 12	v7s: 497	w4f: 56	w7y: 80	y6d: 11
t6s: 1255	t9x: 17	u5t: 30	v12n: 24	v7t: 68	w4g: 47	w8d: 187	y6s: 21
t6t: 198	t9y: 166	u5y: 17	v13h: 11	v7y: 265	w4h: 9956	w8e: 13	y6w: 55
t6x: 17	u10d: 775	u6d: 608	v13t: 20	v8a: 153	w4k: 625	w8g: 108	y7d: 16
t6y: 417	u10e: 20	u6e: 216	v3i: 32	v8d: 92	w4l: 2928	w8l: 16	y7g: 11
t7a: 30	u10g: 69	u6g: 15	v3l: 80	v8e: 152	w4m: 556	w8n: 44	y7r: 40
t7c: 48	u10l: 22	u6l: 87	v3n: 27	v8g: 35	w4n: 2930	w8o: 19	y8f: 160
t7d: 340	u10n: 74	u6n: 32	v4a: 72	v8h: 31	w4p: 68	w8r: 176	y8g: 20
t7e: 126	u10r: 12	u6r: 19	v4d: 20	v8l: 18	w4r: 28	w8s: 160	y8l: 15
t7f: 32	u10s: 23	u6s: 336	v4e: 171	v8n: 37	w4s: 160	w8t: 10	y8n: 10
t7g: 579	u10t: 56	u6t: 45	v4i: 39	v8r: 50	w4t: 4856	w8w: 41	y8t: 11
t7h: 891	u10y: 71	u6y: 26	v4l: 21	v8s: 242	w4y: 16	w8y: 17	y9l: 11
t7l: 104	u11d: 54	u7a: 10	v4n: 136	v8t: 25	w5d: 2652	w9d: 139	y9y: 56
t7n: 101	u11e: 78	u7d: 96	v4s: 60	v8y: 104	w5e: 3284	w9e: 35	z4e: 23
t7o: 49	u11g: 42	u7e: 19	v4t: 65	v9a: 17	w5f: 11	w9g: 81	z4l: 24
t7r: 119	u11s: 41	u7g: 21	v4w: 173	v9e: 86	w5g: 116	w9i: 29	z5m: 12
t7s: 882	u11t: 15	u7l: 37	v4y: 1394	v9g: 20	w5h: 4911	w9k: 20	z6r: 12
t7t: 950	u11y: 111	u7m: 119	v5a: 93	v9i: 12	w5l: 29	w9l: 33	z7s: 10
t7y: 163	u12d: 29	u7n: 97	v5d: 61	v9l: 18	w5n: 664	w9n: 96	z7v: 37
t8a: 23	u12e: 43	u7o: 42	v5e: 630	v9n: 24	w5r: 194	w9r: 37	
t8c: 32	u12s: 17	u7s: 93	v5l: 45	v9r: 12	w5s: 1155	w9s: 72	

B Problem 2

We list here the execution results of our solution for the second problem (movie analytics) tested on the `movies.csv` dataset. Task 1 (duration per country) produces the following results:

Afghanistan: 815	Denmark: 44186	Kosovo: 188	Portugal: 15232
Albania: 646	Dominican Republic: 866	Kuwait: 273	Puerto Rico: 1322
Algeria: 2115	East Germany: 2095	Kyrgyzstan: 400	Qatar: 1036
American Samoa: 247	Ecuador: 893	Laos: 289	Republic of Macedonia: 1515
Angola: 357	Egypt: 3090	Latvia: 2297	Reunion: 38
Argentina: 28813	El Salvador: 167	Lebanon: 1289	Romania: 15739
Armenia: 264	Estonia: 4651	Liberia: 283	Russia: 50757
Aruba: 739	Ethiopia: 537	Libya: 520	Rwanda: 527
Australia: 66068	Faroe Islands: 97	Liechtenstein: 1011	Samoa: 110
Austria: 22925	Federal Republic of Yugoslavia: 2167	Lithuania: 3772	Saudi Arabia: 490
Bahamas: 560	Finland: 48990	Luxembourg: 11976	Senegal: 1462
Bahrain: 87	France: 467279	Macao: 175	Serbia: 4337
Bangladesh: 429	Gabon: 80	Madagascar: 204	Serbia and Montenegro: 1314
Barbados: 92	Georgia: 1846	Malaysia: 1338	Singapore: 4557
Belarus: 427	Germany: 226808	Mali: 220	Slovakia: 2525
Belgium: 54192	Ghana: 655	Malta: 885	Slovenia: 2956
Bermuda: 95	Greece: 20305	Martinique: 103	Somalia: 90
Bhutan: 291	Greenland: 90	Mauritania: 515	South Africa: 14688
Bolivia: 797	Grenada: 79	Mexico: 35560	South Korea: 56245
Bosnia and Herzegovina: 2207	Guatemala: 638	Micronesia: 85	Soviet Union: 51353
Botswana: 295	Guinea: 90	Moldova: 95	Spain: 116651
Brazil: 27333	Haiti: 498	Monaco: 189	Sri Lanka: 318
Bulgaria: 5430	Honduras: 80	Mongolia: 186	Suriname: 105
Burkina Faso: 903	Hong Kong: 66567	Montenegro: 439	Sweden: 68956
Burma: 95	Hungary: 19668	Morocco: 4525	Switzerland: 34516
Cambodia: 894	Iceland: 7839	Namibia: 105	Syria: 447
Cameroon: 615	India: 110215	Nepal: 797	Taiwan: 16618
Canada: 210826	Indonesia: 3931	Netherlands: 47054	Tajikistan: 351
Chad: 493	Iran: 11762	New Zealand: 14561	Tanzania: 441
Chile: 6178	Iraq: 1365	Nicaragua: 183	Thailand: 10189
China: 42089	Ireland: 26506	Nigeria: 494	The Democratic Republic Of Congo: 98
Colombia: 3810	Isle Of Man: 396	North Korea: 508	Trinidad and Tobago: 86
Congo: 88	Israel: 16662	Norway: 26476	Tunisia: 2236
Costa Rica: 431	Italy: 288407	Pakistan: 1728	Turkey: 20068
Croatia: 4340	Jamaica: 807	Palestine: 1530	UK: 492616
Cuba: 2899	Japan: 178558	Panama: 903	USA: 2276142
Cyprus: 948	Jordan: 1313	Papua New Guinea: 295	Uganda: 243
Czech Republic: 15317	Kazakhstan: 2145	Paraguay: 317	Ukraine: 4616
Czechoslovakia: 8434	Kenya: 523	Peru: 2436	
Côte d'Ivoire: 180	Korea: 90	Philippines: 7882	
		Poland: 35543	

United Arab Emirates: 3202	Uzbekistan: 324	Vietnam: 1261	Zaire: 80
Uruguay: 2038	Vanuatu: 100	West Germany: 58480	Zimbabwe: 210
	Venezuela: 2014	Yugoslavia: 8742	

Task 2 (movies per year and genre) produces the following results:

1973_Adventure: 1	1982_Comedy: 1	2002_Documentary: 1	2007_War: 1
1973_Drama: 1	1982_Drama: 1	2002_Music: 1	2008_Action: 1
1973_History: 1	1985_Documentary: 1	2004_Adventure: 1	2008_Crime: 1
1976_Comedy: 2	1985_History: 1	2004_Documentary: 2	2008_Drama: 1
1976_Drama: 1	1988_Comedy: 1	2004_Drama: 1	2009_Documentary: 1
1976_History: 1	1988_Drama: 2	2004_Music: 1	2015_Action: 1
1976_Mystery: 1	1988_Sci-Fi: 1	2004_Romance: 1	2015_Adventure: 1
1976_Romance: 1	1996_Documentary: 1	2006_Animation: 1	2015_Animation: 1
1979_Adventure: 1	1996_Music: 1	2006_Crime: 1	
1979_Crime: 1	2001_Documentary: 1	2006_Drama: 1	
1979_Mystery: 1	2001_Music: 1	2007_Documentary: 1	

C Problem 3

We list here the execution results of our solution for the third problem (movie analytics) tested on the `ecoli.txt` dataset. Task 1 (duration per country) produces the following results:

Afghanistan: 815	Bulgaria: 5430	El Salvador: 167	Indonesia: 3931
Albania: 646	Burkina Faso: 903	Estonia: 4651	Iran: 11762
Algeria: 2115	Burma: 95	Ethiopia: 537	Iraq: 1365
American Samoa: 247	Cambodia: 894	Faroe Islands: 97	Ireland: 26506
Angola: 357	Cameroon: 615	Federal Republic of Yu-	Isle Of Man: 396
Argentina: 28813	Canada: 210826	goslavia: 2167	Israel: 16662
Armenia: 264	Chad: 493	Finland: 48990	Italy: 288407
Aruba: 739	Chile: 6178	France: 467279	Jamaica: 807
Australia: 66068	China: 42089	Gabon: 80	Japan: 178558
Austria: 22925	Colombia: 3810	Georgia: 1846	Jordan: 1313
Bahamas: 560	Congo: 88	Germany: 226808	Kazakhstan: 2145
Bahrain: 87	Costa Rica: 431	Ghana: 655	Kenya: 523
Bangladesh: 429	Croatia: 4340	Greece: 20305	Korea: 90
Barbados: 92	Cuba: 2899	Greenland: 90	Kosovo: 188
Belarus: 427	Cyprus: 948	Grenada: 79	Kuwait: 273
Belgium: 54192	Czech Republic: 15317	Guatemala: 638	Kyrgyzstan: 400
Bermuda: 95	Czechoslovakia: 8434	Guinea: 90	Laos: 289
Bhutan: 291	Côte d'Ivoire: 180	Haiti: 498	Latvia: 2297
Bolivia: 797	Denmark: 44186	Honduras: 80	Lebanon: 1289
Bosnia and Herzegovina: 2207	Dominican Republic: 866	Hong Kong: 66567	Liberia: 283
Botswana: 295	East Germany: 2095	Hungary: 19668	Libya: 520
Brazil: 27333	Ecuador: 893	Iceland: 7839	Liechtenstein: 1011
	Egypt: 3090	India: 110215	Lithuania: 3772

Luxembourg: 11976	North Korea: 508	Senegal: 1462	The Democratic Republic
Macao: 175	Norway: 26476	Serbia: 4337	Of Congo: 98
Madagascar: 204	Pakistan: 1728	Serbia and Montenegro: 1314	Trinidad and Tobago: 86
Malaysia: 1338	Palestine: 1530	Singapore: 4557	Tunisia: 2236
Mali: 220	Panama: 903	Slovakia: 2525	Turkey: 20068
Malta: 885	Papua New Guinea: 295	Slovenia: 2956	UK: 492616
Martinique: 103	Paraguay: 317	Somalia: 90	USA: 2276142
Mauritania: 515	Peru: 2436	South Africa: 14688	Uganda: 243
Mexico: 35560	Philippines: 7882	South Korea: 56245	Ukraine: 4616
Micronesia: 85	Poland: 35543	Soviet Union: 51353	United Arab Emirates: 3202
Moldova: 95	Portugal: 15232	Spain: 116651	Uruguay: 2038
Monaco: 189	Puerto Rico: 1322	Sri Lanka: 318	Uzbekistan: 324
Mongolia: 186	Qatar: 1036	Suriname: 105	Vanuatu: 100
Montenegro: 439	Republic of Macedonia: 1515	Sweden: 68956	Venezuela: 2014
Morocco: 4525	Reunion: 38	Switzerland: 34516	Vietnam: 1261
Namibia: 105	Romania: 15739	Syria: 447	West Germany: 58480
Nepal: 797	Russia: 50757	Taiwan: 16618	Yugoslavia: 8742
Netherlands: 47054	Rwanda: 527	Tajikistan: 351	Zaire: 80
New Zealand: 14561	Samoa: 110	Tanzania: 441	Zimbabwe: 210
Nicaragua: 183	Saudi Arabia: 490	Thailand: 10189	

Task 2 (movies per year and genre) produces the following results:

1973_Adventure: 1	1982_Comedy: 1	2002_Documentary: 1	2007_War: 1
1973_Drama: 1	1982_Drama: 1	2002_Music: 1	2008_Action: 1
1973_History: 1	1985_Documentary: 1	2004_Adventure: 1	2008_Crime: 1
1976_Comedy: 2	1985_History: 1	2004_Documentary: 2	2008_Drama: 1
1976_Drama: 1	1988_Comedy: 1	2004_Drama: 1	2009_Documentary: 1
1976_History: 1	1988_Drama: 2	2004_Music: 1	2015_Action: 1
1976_Mystery: 1	1988_Sci-Fi: 1	2004_Romance: 1	2015_Adventure: 1
1976_Romance: 1	1996_Documentary: 1	2006_Animation: 1	2015_Animation: 1
1979_Adventure: 1	1996_Music: 1	2006_Crime: 1	
1979_Crime: 1	2001_Documentary: 1	2006_Drama: 1	
1979_Mystery: 1	2001_Music: 1	2007_Documentary: 1	

C.1 Problem 4

In the following 4-columned pages, the experimental results of section's 5 problem are presented, as the raw output of the 3rd MapReduce phase execution.

102: 20.772281	110: 16.304225	115: 33.936378	1182: 10.499468
104: 29.609879	111: 42.771938	116: 25.765458000000002	1183: 8.272158999999998
105: 27.598046999999998	112: 9.670065000000001	117: 30.000875	1186: 8.551493
107: 14.800885000000003	113: 28.587929000000003	1180: 9.566474	120: 8.667856
109: 27.163701999999994	114: 17.573652999999997	1181: 8.554887	121: 24.678666000000003

122: 62.881582	221: 15.3660220000000003	289: 20.5074600000000005	348: 13.6963880000000002
124: 23.14891	222: 10.174368	29: 10.500267	349: 12.777321
128: 13.8600000000000001	223: 31.7837630000000004	290: 15.8023840000000002	35: 8.8314610000000001
129: 13.828262	226: 11.464125999999998	292: 34.518677	350: 13.7003560000000001
130: 12.778032	227: 9.910114	293: 42.414617	351: 12.82703
131: 14.548629	228: 26.034533	294: 39.387104	352: 18.913772999999996
132: 13.7801120000000003	229: 17.984762	295: 33.319982999999986	353: 13.8403020000000001
133: 12.777426	23: 8.068862	296: 26.437148	354: 12.026345
134: 17.525341	231: 11.600534	30: 13.1246540000000001	355: 13.8389980000000002
135: 12.8588820000000001	232: 21.044421999999994	300: 16.612497	356: 13.7090900000000002
136: 13.721149	24: 15.9123600000000001	301: 35.5088550000000004	36: 7.7511570000000001
137: 13.6608030000000001	244: 20.7399240000000002	302: 40.46924	361: 13.740885
138: 13.7382540000000001	245: 26.425946999999999	303: 30.324057999999997	37: 9.843972
140: 11.448863	246: 12.422665	304: 34.186174	378: 7.955433
141: 12.826131	247: 12.1988110000000001	306: 12.180608	379: 11.753690999999998
143: 12.803862	249: 19.911965999999996	308: 49.7807830000000014	38: 10.698117
154: 13.7232900000000002	25: 9.950251999999999	309: 21.140049	380: 9.03648
170: 35.833753999999999	250: 20.231632999999995	310: 16.898876	381: 31.255191
174: 39.9352240000000005	251: 35.698739999999999	311: 9.782459	385: 16.697326
175: 42.116396999999999	252: 13.8558760000000002	318: 11.4241890000000002	386: 13.088604999999998
176: 33.780936999999994	254: 8.187458	32: 7.9067320000000001	389: 43.118911000000001
180: 9.111828999999998	255: 27.296618	320: 15.9959230000000003	39: 10.7921260000000001
187: 10.061923	257: 20.167998999999995	322: 12.928547	391: 7.8557420000000001
191: 21.271617	258: 13.3437840000000001	323: 12.95803	40: 7.8420080000000001
192: 22.415023999999995	260: 33.483572999999986	324: 14.1454670000000002	405: 18.636048
193: 24.815881999999995	261: 32.508742999999996	325: 16.820611	406: 30.941025999999999
194: 15.367017	264: 12.784288	326: 9.210550999999999	408: 16.025723
197: 19.721797	265: 17.312394	327: 8.77006	409: 21.124420999999998
198: 20.10406	266: 16.947997	328: 11.502478	410: 19.690642
200: 26.785185	267: 26.829961999999999	329: 9.477534	412: 12.4488180000000001
202: 20.554610999999998	268: 13.1438280000000001	330: 7.819681999999999	413: 21.007549999999995
203: 26.437709999999992	269: 9.54012	331: 12.199909	414: 19.201738999999996
204: 21.1938610000000002	27: 18.650638999999998	333: 17.355712	415: 20.822085
205: 18.054205999999997	270: 12.2007840000000002	334: 16.773555	416: 20.244446999999994
206: 12.843962	272: 27.8627	335: 12.816709	417: 20.336876999999998
208: 18.285626	273: 20.4493260000000003	336: 9.4728200000000002	418: 20.905703999999997
209: 9.478037	274: 21.7819690000000004	337: 16.717836	419: 16.571078
210: 18.463774999999995	275: 18.9216940000000002	338: 14.4356810000000002	420: 14.8337220000000003
211: 26.206232	277: 13.8993770000000001	34: 8.902532	421: 22.127405999999997
213: 8.21817	278: 45.286856999999998	341: 13.7120190000000002	422: 20.207241999999997
214: 19.8097530000000004	279: 36.933221999999994	342: 12.7771680000000001	423: 19.837138999999997
215: 22.834134999999996	280: 36.452380999999999	343: 21.602388999999999	424: 22.204759999999993
216: 22.450407999999996	283: 21.347036	344: 13.763916	425: 20.3222
217: 12.86093	285: 39.845339	345: 9.50445	427: 18.5734230000000002
219: 13.857943	287: 15.863291	346: 13.803495	428: 18.402874999999998
220: 26.279680999999997	288: 24.174710999999995	347: 13.7704250000000001	429: 20.996679999999998

431: 17.247738000000005	537: 15.558095000000002	669: 12.719472000000001	80: 19.708188999999997
432: 7.730989	538: 20.604023999999995	670: 12.759070000000001	81: 27.764969999999987
433: 20.990592000000003	539: 15.567038000000002	671: 18.786868000000002	82: 8.954957
437: 13.467927000000001	540: 10.592666000000001	672: 12.737939000000003	822: 7.824752000000001
438: 22.538098999999995	541: 13.301682	673: 11.354866	827: 26.284926
439: 20.257774999999995	542: 13.708515000000002	674: 12.464726000000002	828: 31.257229999999999
440: 18.640948999999996	544: 8.796812000000001	675: 14.512069	83: 16.828261
441: 22.565751999999993	545: 7.992185	676: 12.633483	830: 12.777338
442: 22.468430999999992	546: 8.689068	677: 15.30606	84: 17.39505
443: 22.236632999999994	547: 17.435207	678: 13.630705	845: 19.799999999999997
444: 12.668758	548: 14.626078000000001	679: 22.209649	846: 23.372872999999995
445: 21.391153999999997	549: 8.902671000000002	680: 26.495342999999999	847: 21.634598999999994
446: 20.229419	550: 15.475738000000002	681: 11.586703	848: 21.599697999999993
447: 23.098830999999993	563: 12.416060000000002	682: 11.780345000000002	849: 20.692736999999997
448: 23.389274999999994	604: 14.693641000000001	683: 10.837639	85: 17.810776
449: 21.425622999999995	605: 14.628854	684: 11.369396	850: 21.486940999999995
450: 20.304112999999994	606: 13.673755000000003	685: 20.680481999999994	851: 20.637819999999998
451: 21.309653999999995	607: 13.823655000000002	686: 28.644553999999992	852: 20.758516999999994
452: 21.574356999999992	608: 13.666609	687: 13.618457000000001	853: 19.714644999999994
453: 21.216737999999992	609: 14.749676000000001	70: 12.576885000000003	854: 20.674989999999998
454: 21.305387999999994	610: 14.726731000000003	702: 15.804968	858: 10.127550999999999
455: 22.443621999999994	611: 12.480203	703: 18.989590999999997	859: 13.744827
456: 18.427684	612: 21.530917999999996	711: 8.769808000000001	86: 9.620828000000001
457: 23.612626999999996	639: 16.345097000000003	72: 13.937429	860: 13.392533000000002
458: 20.137554999999995	64: 11.651773	729: 11.920043	861: 12.716919
468: 16.434241	640: 26.795459999999999	73: 33.317902999999994	862: 14.757012000000001
473: 15.624302000000002	641: 27.955416999999999	739: 14.833940000000002	863: 14.737124000000001
490: 15.118589000000002	642: 22.714243999999994	74: 9.418914	870: 12.684738000000001
492: 29.037201999999994	643: 28.925386	741: 17.572777	871: 12.674369
494: 14.964478999999999	644: 27.951106	743: 16.54965	872: 11.814398
497: 7.779587000000001	645: 9.891883	744: 13.711973000000002	873: 12.763509
505: 22.493396999999998	646: 22.266540999999997	745: 14.759685000000001	874: 10.89
507: 16.502023	647: 17.500691000000003	746: 14.843160000000001	875: 12.642457000000002
508: 13.205226000000001	653: 31.201086999999998	75: 8.17184	876: 12.794634000000002
509: 10.297368000000002	654: 11.212689	76: 35.164774999999998	877: 9.48321
510: 17.989523	655: 15.470210000000002	765: 8.972965	878: 12.762451
511: 14.096020000000003	656: 13.714576000000001	773: 9.301499	879: 13.635698000000001
512: 7.909458999999998	659: 14.530663000000002	778: 9.388625000000001	880: 12.768133
529: 16.491923000000003	660: 14.484713000000001	779: 8.382993	881: 12.801868000000002
530: 20.546952999999999	661: 14.486255000000002	78: 35.074965	882: 10.89
531: 13.581469	663: 14.393031	79: 18.829224	893: 7.887907
532: 18.63035	664: 15.657084000000003	792: 9.650819	894: 9.918199
533: 16.137291	665: 12.470333000000002	793: 21.405161999999994	895: 16.402685000000005
534: 15.585291000000002	666: 15.570483000000001	794: 7.747058000000001	896: 15.014125
535: 13.489995	667: 8.476292	796: 7.746258000000001	901: 10.018733
536: 15.663219000000002	668: 8.400375	797: 20.176615999999996	902: 18.710283

915: 9.631426000000001	927: 18.237036999999997	933: 20.786459999999995	950: 20.797809999999995
917: 9.638277000000002	928: 18.419195	935: 20.676130999999998	951: 19.461768999999997
918: 10.553774	929: 20.752049999999997	936: 21.667931999999997	953: 18.305716999999998
919: 8.431745	930: 13.043696999999998	937: 19.520974999999996	
922: 10.480677000000002	931: 18.611175	941: 25.234674999999996	
926: 20.516713999999997	932: 20.685094999999997	948: 16.893155	