

Informe Tarea 2

Luis Ramírez Donoso

La tarea Nº 2 de Sistemas Recomendadores se basa en la implementación de un algoritmo que pueda recomendar artículos en base a contenidos, es decir, utilizar Content Based Recommender .

La tarea en particular cuenta de 4 actividades.

1.- Leer en el data

Consiste en extender 2 clases, agregando dos columnas del archivo de entrada. Los campos son título y descripción. Las clases involucradas en esta modificación son common/Bookmark y file/BookmarkReader.

Dentro de common/Bookmark

```
public class Bookmark implements Comparable<Bookmark> {  
  
    private int userID;  
    private int resID;  
    private String timestamp;  
    private List<Integer> tags;  
  
    private double rating;  
    private List<Integer> categories;  
  
    private String title;  
    private String description;  
  
    public Bookmark(int userID, int wikiID, String timestamp,  
String title,String description) {  
        this.userID = userID;  
        this.resID = wikiID;  
        this.timestamp = timestamp;  
  
        this.title = title;  
        this.description = description;  
  
        this.rating = -2.0;  
        this.categories = new ArrayList<Integer>();  
    }  
}
```

```

        this.tags = new ArrayList<Integer>();
    }

    @Override
    public int compareTo(Bookmark data) {
        //return (Long.parseLong(getTimestamp()) <=
        Long.parseLong(data.getTimestamp()) ? - 1 : 1);
        if (this.userID < data.getUserID()) {
            return -1;
        } else if (this.userID > data.userID) {
            return 1;
        } else {
            if (!this.timestamp.isEmpty() &&
            !data.timestamp.isEmpty()) {
                if (Long.parseLong(this.timestamp) <
                Long.parseLong(data.timestamp)) { // <
                    return -1;
                } else if (Long.parseLong(this.timestamp)
                > Long.parseLong(data.timestamp)) { // >
                    return 1;
                }
            }
        }
        return 0;
    }

    // Getter -----
    -----

    public void setUserID(int userID) {
        this.userID = userID;
    }

    public int getUserID() {
        return this.userID;
    }

    public void setWikiID(int wikiID) {
        this.resID = wikiID;
    }

    public int getWikiID() {
        return this.resID;
    }

```

```

    }

    public String getTimestamp() {
        return this.timestamp;
    }

    public String getTitle() {
        return this.title;
    }

    public String getDescription() {
        return this.description;
    }

    public double getRating() {
        return this.rating;
    }

    public void setRating(double rating) {
        this.rating = rating;
    }

    public List<Integer> getCategories() {
        return this.categories;
    }

    public void setTags(List<Integer> tags) {
        this.tags = tags;
    }

    public List<Integer> getTags() {
        return this.tags;
    }
}

```

Dentro de file/BookmarkReader

```

processUserData(userID, userData, tags, categories, wikiID,
title, description);
    // reset userdata
    userID = lineParts[0].replace("\\\"", "");
    wikiID = lineParts[1].replace("\\\"", "");
    timestamp = lineParts[2].replace("\\\"", "");
    userData = new Bookmark(-1, -1, timestamp,

```

```

title, description);
        categories.clear();
        tags.clear();
        for (String tag :
lineParts[3].replace("\\", "").split(",")) {
            if (!tag.isEmpty()) {
                String stemmedTag =
tag.toLowerCase();

                if (this.stemmer != null) {

                    this.stemmer.setCurrent(stemmedTag);
                    this.stemmer.stem();
                    stemmedTag =
this.stemmer.getCurrent();
                }
                tags.add(stemmedTag);
            }
        }
        if (lineParts.length > 4) { // are there
categories
            for (String cat :
lineParts[4].replace("\\", "").split(",")) {
                if (!cat.isEmpty()) {
                    if (cat.contains("_")) {

                        categories.add(cat.substring(0,
cat.indexOf("_")).toLowerCase());
                    } else {

                        categories.add(cat.toLowerCase());
                    }
                }
            }
        }
        if (lineParts.length > 5) { // is there a
rating?
            try {

                userData.setRating(Double.parseDouble(lineParts[5].replace(
"\", "")));
            } catch (Exception e) { /* do nothing
*/ }
        }
        title = lineParts[6].replace("\\", "");

```

```

description = lineParts[7].replace("\"",
");
// TODO -----
// extend common/Bookmark class with fields
for title (= lineParts[6]) and description (= lineParts[7])
    }
    processUserData(userID, userData, tags,
categories, wikiID, title, description); // last user
    br.close();
    return true;
} catch (Exception e) {
    System.out.println("ERROR");
    e.printStackTrace();
}
return false;
}

```

2.- Implemetar un Tag recomendador basado en contenido

Para esta tarea fue muy difícil lograr implementar. Se realizó un estudio del lenguaje JAVA para entender su sintaxis y estructura. Luego se realizó una investigación de sistemas similares que realizaran algún cálculo para lograr recomendar ítem en base a contenido. Fue muy difícil encontrar, pero lo más difícil fue lograr interpretar para incluirlo en el código de la clase a la cual se debía modificar.

La clase que se debía tocar es processing/ContentBasedCalculator y el método dentro de esta clase es getRankedTagList.

Se logró implementar un cálculo de comparación como se presenta en el siguiente código de la clase:

```

public class ContentBasedCalculator {

    private final static int REC_LIMIT = 10;

    private BookmarkReader reader;
    private List<Bookmark> trainList;

    private List<Map<Integer, Double>> userMaps;
    private List<Map<Integer, Double>> tagRecencies;
    private List<Map<Integer, Double>> resMaps;

```

```

    public ContentBasedCalculator(BookmarkReader reader, int
trainSize) {
        this.reader = reader;

        // TODO: use this data for recommendations
        this.trainList =
this.reader.getBookmarks().subList(0, trainSize);
    }

    public Map<Integer, Double> getRankedTagList(int userID,
int resID) {

        Map<Integer, Double> resultMap = new
LinkedHashMap<Integer, Double>();
        this.tagRecencies = new ArrayList<Map<Integer,
Double>>();
        // TODO: calculate your recommendations here and
return the top-10 (=REC_LIMIT) tags with probability value
        // have also a look on the other calculator classes!

        if (this.userMaps != null && userID <
this.userMaps.size()) {
            Map<Integer, Double> userMap =
this.userMaps.get(userID);
            Map<Integer, Double> recMap =
this.tagRecencies.get(userID);
            for (Map.Entry<Integer, Double> entry :
userMap.entrySet()) {
                resultMap.put(entry.getKey(),
entry.getValue().doubleValue() *
recMap.get(entry.getKey()).doubleValue());
            }
        }
        if (this.resMaps != null && resID <
this.resMaps.size()) {
            Map<Integer, Double> resMap =
this.resMaps.get(resID);
            for (Map.Entry<Integer, Double> entry :
resMap.entrySet()) {
                Double val =
resultMap.get(entry.getKey());
                resultMap.put(entry.getKey(), val ==
null ? entry.getValue().doubleValue() : val.doubleValue() +

```

```

entry.getValue().doubleValue());
        }
    }

    Map<Integer, Double> returnMap = new
LinkedHashMap<Integer, Double>();
    Map<Integer, Double> sortedResultMap = new
TreeMap<Integer, Double>(new DoubleMapComparator(resultMap));
    sortedResultMap.putAll(resultMap);
    int count = 0;
    for (Map.Entry<Integer, Double> entry :
sortedResultMap.entrySet()) {
        if (count++ < 10) {
            returnMap.put(entry.getKey(),
entry.getValue());
        } else {
            break;
        }
    }
    return returnMap;

```

En ésta base del código comentado, se deja en evidencia el cálculo de la recomendación en base a la estrategia utilizada TF-IDF, utilizando el coseno del item y del perfil del usuario.

//Ahora calculo en base a TF-IDF

```

        //DecimalFormat df = new DecimalFormat("0.0000");

        // Obtener el perfil del usuario con el "Me
Gusta" para cada etiqueta

        //SparseVector userVector =
makeUserVector(userID);

        // Loop sobre cada elemento requerido y su
puntuación

        // El dominio del vector de salida es lo que se
conseguirá

```

```

        //for (VectorEntry e:
resID.fast(VectorEntry.State.EITHER))
        //{
            // El Score del item representado por e.
            // Obtener el vector del item para este
item en particular

            // SparseVector iv =
model.getItemVector(e.getKey());
            // Se calcula el coseno del item y el
perfil del usuario
            //Se guarda en el vector de salida

            // double numerator = iv.dot(userVector);
            // double denominator = iv.norm() *
userVector.norm();
            // double cosineSimilarity = numerator /
denominator;
            // try
            // {
            //     double rounded =
df.parse(df.format(cosineSimilarity)).doubleValue();
            //     logger.info("Cos similarity for item
{} and user {} is {}", e.getKey(), userID, rounded);

            //     resID.set(e.getKey(), rounded);
            // }
            // catch (ParseException e1)
            // {
            //     logger.error("Error while parsing
double", e1);
            // }
        // }
    }
}
// -----
-----
-----

```

```

    public static List<Map<Integer, Double>>
startContentBasedCreation(BookmarkReader reader, int sampleSize)
{
    int size = reader.getBookmarks().size();
    int trainSize = size - sampleSize;

```



```

        ContentBasedCalculator calculator = new
ContentBasedCalculator(reader, trainSize);
        List<Map<Integer, Double>> results = new
ArrayList<Map<Integer, Double>>();
        if (trainSize == size) {
            trainSize = 0;
        }

        for (int i = trainSize; i < size; i++) { // the test-
set
            Bookmark data = reader.getBookmarks().get(i);
            Map<Integer, Double> map =
calculator.getRankedTagList(data.getUserID(), data.getWikiID());
            results.add(map);
        }
        return results;
    }

    public static void predictSample(String filename, int
trainSize, int sampleSize) {

        BookmarkReader reader = new BookmarkReader(trainSize,
false);
        reader.readFile(filename);

        List<Map<Integer, Double>> modelValues =
startContentBasedCreation(reader, sampleSize);

        List<int[]> predictionValues = new
ArrayList<int[]>();
        for (int i = 0; i < modelValues.size(); i++) {
            Map<Integer, Double> modelVal =
modelValues.get(i);

            predictionValues.add(Ints.toArray(modelVal.keySet()));
        }
        String suffix = "_cb";

        reader.setUserLines(reader.getBookmarks().subList(trainSize
, reader.getBookmarks().size()));
        PredictionFileWriter writer = new
PredictionFileWriter(reader, predictionValues);
        String outputFile = filename + suffix;
        writer.writeFile(outputFile);
    }
}

```

```
}  
}
```

3.- Evaluación del Tag Recomendador

El resultado del tag implementado es posible comprobar su funcionamiento al editar en el archivo pipeline la línea de código asociada a la ejecución del método.

```
// TODO: just execute to test your recommender - results can be  
found in metrics/bib_core  
startContentBasedCalculator("bib_core",  
"bib_core/bib_sample");
```

El resultado es depositado queda registrado en la ruta data/metrics/bib core y el archivo TXT se llama “cb_metrics_avg.txt”.

El resultado se arroja como sigue:

```
0-3909|2, 3, 353, 335, 396|  
1-570|45, 462, 280, 466, 634, 467, 40|  
2-174|73, 1254, 472|  
3-5|459, 462, 460, 0, 22|  
4-314|107, 462, 468, 40|  
5-1662|1886, 13|  
6-124|409|  
7-1889|695, 696, 52, 697|  
8-430|698|  
9-1016|247, 482, 704|  
10-364|85, 102, 5, 6, 14, 760, 322, 761, 528, 771, 772, 529, 762|  
11-7|8, 87, 25|  
12-1238|775, 463|  
13-1438|14, 529|  
14-1262|616, 339, 354, 484, 820, 464|  
15-1149|446, 47|  
16-1490|1210|  
17-69|784, 45, 785, 225, 254, 29, 778, 786, 787, 83, 788, 170,  
372, 789, 791, 717, 128|  
18-808|793, 128|  
19-1936|18, 79|  
20-4116|1173, 189|
```

21-3508|957|
22-109|6|
23-271|254, 42|
24-233|59, 369, 246, 370, 371, 376, 79, 372, 47, 373, 374, 375,
49|
25-3785|2094|
26-2493|963, 736|
27-1057|1003, 641, 484, 655|
28-726|730, 1031|
29-3911|107, 595, 2551|
30-870|869, 85, 596, 1035, 68|
31-484|1040, 611, 602|
32-1209|141, 247|
33-857|793, 857|
34-1904|242, 863, 23|
35-1232|367, 308|
36-217|355, 122, 246, 35, 356, 357|
37-2082|721, 206, 397, 5|
38-1542|305, 522|
39-1187|863, 91|
40-722|1085, 1101, 707|
41-656|282, 353|
42-2087|107, 1121, 542, 47, 5, 49|
43-2472|715, 1835, 806, 1837|
44-1985|654|
45-1770|247, 1060, 939|
46-801|582, 1183|
47-212|343, 348, 85, 154, 2567, 1185, 147, 970, 64, 422, 296,
1190, 22|
48-233|47, 353|
49-1422|172, 672, 4, 5|
50-526|247, 621|
51-602|1197|
52-4083|2553, 467|
53-14|114, 1200, 609, 49|
54-83|567|
55-1289|141, 45, 733, 242, 167|
56-653|14, 22|
57-1477|1202|
58-1189|1077, 282, 342, 45, 32|
59-109|645|
60-2661|179|
61-1222|707, 217, 1893, 1102, 110|
62-2076|117, 107, 83, 79, 47, 68, 49|

63-3503|1053|
64-1866|898, 706, 1543, 717|
65-1542|1068, 305, 936|
66-361|61, 1689|
67-456|79, 84, 278|
68-632|106, 305, 936|
69-407|50, 51, 487, 1074, 5|
70-3875|239, 249, 14, 1678|
71-607|62, 1266, 560, 1268, 311|
72-1370|271, 1274, 225, 963, 812, 500, 273, 1150, 1181|
73-1966|1285, 1281, 1284, 1280, 1283, 396|
74-4039|1286|
75-2467|1368, 1588, 1292, 1294|
76-918|687|
77-3340|426, 388, 1332, 553|
78-984|1207, 768, 85, 198|
79-381|117, 79, 213, 357, 457|
80-1028|41, 39, 5, 6|
81-1527|141, 45, 254, 1426|
82-110|50, 51, 52, 175|
83-1105|1339, 1337, 1023, 13|
84-1475|1341, 188, 22|
85-2443|1811, 582, 703, 655, 1534|
86-268|49|
87-1641|254, 604, 556, 1219, 26, 42, 1259, 124|
88-323|65, 84, 40|
89-2531|498|
90-3948|239, 324, 855|
91-427|106, 1077, 1060, 1201|
92-3302|1240|
93-1744|354|
94-3884|117, 2, 1392, 6, 1393, 1212, 756, 5, 660, 746|
95-1145|6, 1099, 5, 418|
96-3948|401, 298, 239, 103, 85, 962, 2977|
97-388|30, 32|
98-780|247, 1397, 645, 14|
99-1953|1398|
100-1424|1401, 323, 5, 6|
101-2402|181|
102-2027|1416, 60, 774, 206, 1419, 1413|
103-2015|342, 38|
104-804|748, 1437, 339|
105-3882|1440, 107|
106-460|432, 588, 589|

107-2486|1448, 1762|
108-1282|565|
109-1210|1080, 107, 952, 1462, 1091, 14, 1460, 1466, 1092, 1067|
110-804|1469|
111-1360|1580, 1144|
112-1541|137, 2, 935, 1475, 6, 305, 111, 98, 936|
113-3331|231, 14, 898|
114-816|340, 810, 32|
115-934|1510|
116-971|903|
117-1597|114, 188, 47, 49|
118-845|845|
119-407|1797, 52, 557, 5|
120-1583|1526, 133|
121-1571|64, 6|
122-205|21, 64, 155|
123-850|50, 142, 14|
124-639|59, 499, 49|
125-1873|1900, 63, 32, 26|
126-719|241, 13, 1060|
127-4161|1037, 1267, 1032|
128-256|725|
129-615|673|
130-948|903, 266|
131-385|94, 419|
132-3564|107, 1557, 1564, 1099, 498, 1566|
133-1914|835, 2801|
134-1255|1547, 1347|
135-938|915|
136-1421|1582, 645, 939|
137-1188|93|
138-396|29, 337, 313|
139-4302|3642, 1352, 3643|
140-448|49, 1050, 1382, 79, 388|
141-1193|1595|
142-4054|282, 620, 2621, 2607, 1942, 1157|
143-4352|8, 685|
144-1147|774, 206|
145-1023|72, 1406, 98, 221|
146-1546|1283|
147-1517|208|
148-1704|112, 14|
149-1777|29, 85|
150-804|45|

151-3508|1613, 1604, 85, 649, 402, 35, 103|
152-2028|1606, 466, 1610, 1608, 1029|
153-435|2151, 47, 68|
154-3457|107, 1606, 253, 1608, 1425|
155-525|107, 1631, 1090, 47, 49|
156-1344|1649, 135, 47|
157-66|123, 311, 20, 366, 278, 21, 64, 127, 124, 22|
158-2044|6|
159-2094|193, 1717, 177|
160-1607|188, 189|
161-1285|335|
162-222|85, 726, 14, 361|
163-1921|341, 154, 1665, 40, 344|
164-1384|485, 206, 1666, 105, 1236|
165-3566|128|
166-829|803|
167-3875|239, 249, 14, 1678, 936|
168-156|171, 94, 41, 206, 1706|
169-1216|1714, 1081, 1713, 1466|
170-1097|715, 852, 39, 206, 645, 1715, 1252, 5|
171-868|400, 1621, 47|
172-168|6, 5|
173-471|546|
174-2369|142, 342|
175-2378|1755, 1751, 1347|
176-2343|239, 659|
177-2182|1111, 1757|
178-2362|1755, 1761|
179-3319|810, 582|
180-3819|1590, 976|
181-890|50, 228, 713, 879, 887, 1426|
182-2412|354, 1099, 237|
183-895|659, 1787|
184-4352|1790|
185-2394|107, 595, 869, 1791, 2296, 455, 843, 1036, 47|
186-1891|246|
187-651|52, 308|
188-2059|764, 1267, 567, 1800, 1801, 1667, 706|
189-2425|239, 446, 1789, 1210|
190-135|62, 1762, 638, 409|
191-161|1803, 781, 217, 240, 241, 13, 54|
192-551|206|
193-4159|1558|
194-29|141, 79, 1808, 42|

195-3532|141|
196-2459|1821, 1207, 1820|
197-1968|52|
198-4353|117, 306, 971, 84, 64|
199-4353|214, 971, 405, 84, 64|
200-245|120, 88, 140|
201-1561|1833, 550, 1298, 1294|
202-4354|1434, 1590, 836, 2634|
203-1989|45, 342, 35, 654, 2382, 68|
204-2488|1448, 1845|
205-3599|1450, 113|
206-758|378, 1808, 582|
207-17|14, 49|
208-2486|1344|
209-2055|64, 596, 29, 858, 335, 64, 250, 64|
210-341|50, 51, 493|
211-234|377, 378, 254, 1854|
212-3889|1828, 1129, 1855, 149|
213-1773|254|
214-3122|669|
215-2983|732, 582, 1862, 470|
216-1461|1554, 666, 556, 32, 365, 1043, 632|
217-1601|48, 1667|
218-2398|1077, 219, 114, 1769, 47, 49|
219-404|963, 1878, 404|
220-4160|654, 1881, 1447|
221-3329|1884, 1650, 1173|
222-3880|153, 11|
223-1421|1892, 6, 939, 5|
224-3574|935, 60, 68, 1897, 91|
225-3837|1939, 94|
226-694|638, 3610, 3609|
227-61|60, 29, 1898, 372|
228-3876|1930, 436, 46|
229-2082|206, 5|
230-3530|1270|
231-1467|794, 1196, 38, 47|
232-156|711, 206, 40, 397, 5|
233-2582|1909|
234-65|122, 600, 2513, 54|
235-1402|939|
236-4352|685, 713, 1914|
237-1644|370|
238-1778|707, 1472, 1917|

239-347|14|
240-1560|889, 6, 5|
241-640|50, 85, 39, 103|
242-2821|1933|
243-116|342, 1629|
244-3822|736|
245-3312|1940, 1942, 654, 1938|
246-3311|1940, 1941, 179, 1938|
247-3310|179, 1938|
248-3267|32, 32, 824, 824, 1934, 1936, 1935|
249-4160|2061, 466, 449|
250-2933|965|
251-1625|590|
252-1531|335|
253-781|713, 1799|
254-1498|319, 582, 189|
255-751|240|
256-1790|1957, 1956|
257-2473|3500, 35, 3622|
258-3448|2056, 2057, 1971, 2029|
259-3430|1971, 2012|
260-3370|1969, 1970|
261-661|85, 32, 1988|
262-3787|98|
263-837|32, 237|
264-3427|179|
265-3357|1864|
266-1434|107|
267-1416|2077, 14|
268-169|369|
269-1629|154|
270-1517|794, 198, 1420, 1697|
271-3476|2080|
272-3474|2082, 2080|
273-3474|2082, 2080|
274-234|469, 2083, 42|
275-2803|248, 1710, 1747|
276-3476|198, 2080|
277-269|6, 5|
278-2406|595, 35|
279-276|106, 62, 107, 410, 2, 4, 122, 408, 523, 188, 477, 49|
280-1381|208, 2469|
281-1857|175|
282-2575|188|

283-490|87|
284-1721|29, 1009, 254|
285-3312|1450, 35, 1942, 654, 1394, 2850|
286-3817|596, 46, 2542|
287-533|45|
288-1508|826|
289-3470|1761|
290-1966|117, 2107, 1281, 2108, 1283|
291-2069|172, 45, 29, 463, 683, 1347, 1059, 73, 103, 179, 5|
292-2090|1420|
293-1788|1114|
294-3459|35, 1366, 5|
295-1805|2, 79, 113|
296-3874|359|
297-2432|1457, 3, 57, 15, 1747|
298-110|1094|
299-3855|2117, 304|
300-1297|902, 217, 2118, 2119, 32|
301-1319|1277|
302-71|1631|
303-2063|659, 313, 660|
304-3873|171, 2096|
305-3531|2132, 2133, 179|
306-1061|239, 999, 582, 1000, 813|
307-3229|110|
308-4353|2, 971|
309-2101|64, 1111, 142, 342, 1724|
310-878|247|
311-3808|93|
312-3504|206, 2146|
313-3801|1946|
314-1682|49|
315-502|1145|
316-2512|1009, 1620|
317-2023|2117, 304|
318-1915|469, 436, 2151|
319-1100|906, 527|
320-806|801|
321-1952|1597|
322-187|1094|
323-2058|195, 1664|
324-1943|905|
325-65|2513|
326-810|826|

327-2520|93|
328-4159|102, 2160, 2161, 2164|
329-2868|2172|
330-1673|645, 978, 14|
331-428|50, 126, 511, 246|
332-388|32, 717|
333-68|125, 122, 22|
334-2481|402, 38|
335-3579|79|
336-3547|1610, 1029|
337-2361|1752, 768, 1755, 1751, 2200, 1347|
338-3578|1610, 2201|
339-26|1161|
340-2829|2311|
341-119|50, 128, 713, 47|
342-420|2206, 48, 65, 84|
343-1023|2211|
344-1832|228, 2212|
345-3663|2213, 335, 301, 158, 1129, 654|
346-3612|2213, 574, 2068|
347-3588|2213, 2238, 501|
348-3645|29, 144|
349-3708|2362, 2213, 2363, 2364, 2365, 2366, 2353|
350-3765|67, 2213, 1359, 1141, 23|
351-3692|2213, 1229, 2235|
352-911|505, 902, 903, 266|
353-3628|1915, 2213, 512, 2466, 2126|
354-3679|2213, 1084, 548|
355-3662|2492, 2459, 2213|
356-3737|2231, 2422, 2213|
357-1405|886, 2213, 144|
358-3592|2213, 72, 2322|
359-3754|85, 2463|
360-3737|2458, 2213, 103|
361-2479|35|
362-1928|2, 454, 2106, 313, 35, 717|
363-276|2213, 408, 1129, 49|
364-2453|1346, 35, 2518|
365-1154|1592|
366-3767|2213|
367-783|432, 2521, 759, 582, 755, 163|
368-966|841|
369-3324|2523|
370-1252|1410|

371-3825|154, 568, 898, 512, 149|
372-3784|1848|
373-3455|295, 2530, 2167, 2163, 2164, 2160, 2166|
374-725|63, 2531|
375-4355|1854, 14, 2532, 1236|
376-2583|835, 1795|
377-3789|216, 345|
378-1889|52, 2008, 501|
379-103|45, 254, 64, 26, 167, 93|
380-2072|1664, 5|
381-9|34, 6, 24, 36|
382-798|1485, 1482, 1514|
383-1146|968, 1425|
384-3811|2544, 2545, 968|
385-4045|2549|
386-1817|1170, 79, 163, 188|
387-2450|191|
388-3826|141, 1239, 242, 253, 1838, 236, 298, 35|
389-1886|794, 52|
390-2364|1754|
391-2479|35|
392-1065|1006|
393-458|2, 6, 906, 250, 259|
394-3871|687, 1047, 511|
395-2000|715, 32|
396-4077|29, 339, 2553|
397-4116|500, 1173, 189|
398-3847|179|
399-3853|484, 2558|
400-3852|2561|
401-3322|2570, 81|
402-3859|141, 1696, 341, 280, 2106, 2565, 1346, 1620|
403-2425|659, 1789|
404-739|2568|
405-981|50, 29, 85, 144, 361|
406-3915|107|
407-3915|2574|
408-4355|2533, 247, 2532|
409-855|853, 851, 665|
410-3539|2436, 2268, 1129|
411-190|1129|
412-1353|1915, 1776, 978, 14|
413-821|35|
414-3789|216, 768|

415-2368|117, 92, 1734|
416-1033|2579|
417-59|1094|
418-1740|2435, 2580, 113|
419-2096|154, 32, 1722, 47, 221|
420-1707|1016|
421-1934|171|
422-3578|228, 616, 467|
423-1615|1060|
424-286|107|
425-3502|2591, 2106, 1096, 1079|
426-3319|810, 1405, 793|
427-2049|2593|
428-3573|107|
429-1731|280|
430-109|92, 6, 5|
431-3796|1094|
432-1365|2597|
433-3594|877, 1859|
434-827|2773, 846, 820|
435-2085|1188|
436-1335|246, 46, 293|
437-1832|228, 1610, 1394|
438-4040|154, 233, 892, 2755|
439-242|191, 192, 40, 64|
440-225|1145|
441-3453|282, 620, 2621, 207, 807|
442-1013|2627, 1678|
443-4356|107, 895, 2632|
444-4354|1434, 2634|
445-3496|2100|
446-957|794, 239, 323, 241, 568|
447-2393|47|
448-971|1094|
449-2787|2, 1578|
450-764|243, 557|
451-2485|1448|
452-3827|2551, 627, 46, 47|
453-3909|1589|
454-3913|2638|
455-3860|82|
456-122|107, 287, 348, 185, 186, 289|
457-3865|2642|
458-2566|85, 2122|

459-3920|184, 107, 1420|
460-3481|250, 477, 49|
461-3877|556, 2582|
462-2005|645|
463-1329|634, 15|
464-1859|50, 51, 638, 1540|
465-560|32, 823|
466-546|582, 1829, 1087, 1667, 113, 1060|
467-4356|107, 1651, 286|
468-539|1094|
469-3926|556, 2536, 906, 171, 2192|
470-873|864, 242, 47|
471-550|247, 65, 254, 113|
472-1392|1094|
473-3873|6, 746|
474-3479|49, 463, 47|
475-109|6, 2658, 5|
476-530|1408, 101, 2659, 47|
477-1944|1896|
478-250|2590, 93|
479-1601|48, 986, 1656, 84, 457|
480-2460|60|
481-3908|107|
482-661|85|
483-4357|107, 2665, 280, 254, 40, 64, 93|
484-2455|1825, 1376|
485-1011|670, 630|
486-1893|905, 311, 1552|
487-2030|2575|
488-569|107, 942, 47|
489-1215|917, 15, 47|
490-2551|13|
491-3942|92, 26|
492-1245|815|
493-350|1427|
494-1562|1300|
495-3950|3690, 487|
496-1456|2313, 14, 47|
497-368|1094|
498-1961|228, 1046, 94, 863, 498, 1566|
499-3968|2739|
500-4038|181, 2741, 2743|
501-4028|117, 2741, 2748, 2746, 2742, 3641, 2749|
502-5|222, 2179|

503-361|45, 61, 63, 1427|
504-557|599, 547, 47, 5|
505-234|153, 254, 1357|
506-3858|154, 2567, 2756, 64|
507-4044|184, 2757, 2759|
508-4043|2758, 2757, 604|
509-1732|65|
510-1288|2650|
511-378|79, 406|
512-4045|107, 511|
513-262|106, 2, 29, 63, 144, 246, 408|
514-1796|1737, 102|
515-1023|2762, 2763|
516-311|2130|
517-349|2764, 725|
518-244|171, 239|
519-1640|736, 93|
520-3500|403, 52|
521-3952|655|
522-814|2767|
523-2007|794, 2106|
524-1564|1306, 1296, 1307, 1297, 1294|
525-4052|107, 6, 1294|
526-3537|130, 6, 14, 5, 1157|
527-140|203, 45, 194, 461, 26|
528-4052|107, 1293|
529-3466|184, 2771, 2074, 939|
530-242|32, 582, 14, 105, 149|
531-14|49|
532-3568|2772, 831, 611, 819|
533-180|14, 113|
534-1668|645|
535-134|60, 1390, 655, 2829|
536-4050|2362, 1025, 654, 2058|
537-1502|973, 494|
538-113|29, 1849, 178, 180, 179, 143, 47|
539-109|6, 645, 14|
540-2082|163, 1200, 392|
541-167|2776|
542-4058|2200, 2777|
543-2420|731|
544-2573|1180, 18, 48, 206|
545-1815|184, 1523, 29, 79, 213|
546-2632|258, 2, 1356, 29, 859, 340, 1814, 179, 1727, 611, 490,

1756, 686, 313, 660|
547-2610|85, 1921, 2024, 2021|
548-767|2778, 2780, 2779|
549-280|154, 40, 296, 425, 22|
550-235|206|
551-4300|2781, 2782|
552-2489|1854|
553-4355|1854, 14|
554-458|6, 906, 1854, 259|
555-3498|2100, 2103|
556-4357|107, 45, 2665, 41, 64, 26, 138|
557-1496|239, 246, 47|
558-4115|1852, 2785, 1853, 2786, 2788, 2787, 1852, 2789|
559-103|469, 1947, 1948|
560-1434|462, 1182, 1339|
561-4118|281, 868, 2793|
562-3859|696, 2117, 304|
563-4320|2803, 2812, 2804, 2805, 2807|
564-978|179, 266|
565-1911|1557, 634|
566-3852|2561, 1200|
567-1214|219, 68, 2058|
568-541|535, 32, 14|
569-3876|2819, 1763, 2257|
570-4142|1094|
571-1052|1403, 487, 994, 3652|
572-900|961|
573-2728|1094|
574-622|677, 2549, 514|
575-4149|1094|
576-4299|8, 2018, 237|
577-1016|271, 225, 812, 85, 247, 273, 6, 40, 266, 5, 717|
578-2082|582, 2822, 2823, 1074|
579-1187|1094|
580-3910|1094|
581-2825|490|
582-4354|1448, 2634|
583-1318|60, 224, 1130, 213, 2825, 357|
584-3472|462, 436, 213, 345|
585-3920|184, 2649, 1059|
586-104|1240|
587-3472|417|
588-1811|519, 634, 64|
589-149|1030|

590-2525|498, 2519|
591-1035|85, 45, 958, 808|
592-763|2831, 735, 554, 553|
593-149|794, 2094, 47|
594-3910|663|
595-1788|1094|
596-3770|2609|
597-2033|715, 282, 253, 2607, 2116, 376|
598-1578|45, 114, 47, 49|
599-242|489, 262|
600-1140|378, 1808, 43|
601-3930|1094|
602-2597|128|
603-388|484|
604-681|94, 335, 841, 1201|
605-4200|1358, 3028, 3029, 1943, 2002, 3030, 3031, 3032, 2857,
3033, 3034, 3035, 3036|
606-4282|1448, 595, 3059, 1943, 1599, 3022, 2982, 3606, 2634,
1598, 3038, 3024|
607-2131|3562, 354|
608-4268|2979, 3019, 1943, 3536, 3537, 3538|
609-4206|1448, 3059, 1943, 1599, 3022, 3025, 2634, 3024, 3060|
610-571|789|
611-3830|365, 149|
612-527|163, 395, 5|
613-124|29, 154, 144, 84|
614-3883|3611, 376|
615-1337|246|
616-4299|107, 3613|
617-4081|3616, 207|
618-750|1094|
619-9|36|
620-4156|213, 2790|
621-2750|1937, 360, 813|
622-1758|683, 32, 880, 848|
623-398|644, 1667|
624-4081|3615|
625-1023|3633, 1001, 3634, 3617, 3618, 3619|
626-1000|79|
627-793|432, 588, 589|
628-4303|462, 2746|
629-697|1667|
630-823|50, 47, 49|
631-1943|3645|

632-1353|14|
633-623|313, 189|
634-4310|1094|
635-3815|254|
636-812|1095|
637-3931|154, 528|
638-307|59, 92, 145|
639-347|3648, 1714, 1100, 278, 2463|
640-1505|59, 794, 114|
641-3360|224, 1132, 1130, 213, 357, 221|
642-620|45, 159, 160, 47, 49|
643-3360|224, 1130, 689, 213, 2825, 426|
644-189|1181, 3653|
645-330|79, 163, 1384, 1688, 1238|
646-2492|239, 378, 15|
647-2506|1040, 745, 156, 1808|
648-1944|1094|
649-2576|6, 13, 926, 676, 1684|
650-1408|144, 1445, 2257|
651-777|889, 914, 6, 755|
652-3908|224, 440, 85, 3656, 308, 528|
653-770|3657, 1148|
654-1023|2773, 1431, 540|
655-973|517|
656-837|298, 207, 237|
657-1370|2097|
658-335|22|
659-22|2, 3, 638, 6, 77, 259|
660-115|3674|
661-3886|1094|
662-342|154, 270, 108, 40, 425|
663-3802|560, 64, 26, 167, 1946|
664-4328|107|
665-4327|107|
666-4328|1240|
667-3807|2540, 41, 254, 2228, 138|
668-4351|462, 213, 89, 357|
669-4357|107, 2665, 41, 64, 26|
670-434|239|
671-398|3677|
672-4351|462, 2824, 213, 89, 357|
673-4339|45, 6, 570, 3681, 31, 3679, 3683|
674-1853|1094|
675-4315|3670, 3294|

676-4339|107, 45, 462, 6, 570, 3681, 31, 3679, 3683|
677-3319|1123, 1059, 820|
678-1057|32, 641, 30, 1128|
679-1797|1094|
680-1601|1094|
681-2333|142, 342, 848, 15|
682-4303|107|
683-253|29, 277|
684-4097|29, 2553, 620, 207, 2361|
685-3835|1763, 207, 2612|
686-3937|107, 1807|
687-793|2114, 1389|
688-4358|107, 3684|
689-4356|107, 2631, 2632|
690-4330|107, 224, 6, 1158, 3685, 1580|
691-4347|107|
692-4348|341, 85, 1664, 279|
693-9|36|
694-1872|474|
695-292|10, 12, 430, 26|
696-4047|193, 1130, 221|
697-1944|1094|
698-4334|107|
699-0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0
700-0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0
701-0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0
702-0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0
703-0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0
704-0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0
705-0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0
706-0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0
707-0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0
708-0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0