ej_2

June 15, 2021

[3]:
```
import findspark
findspark.init()
findspark.find()
import pyspark
findspark.find()
from pyspark import SparkContext, SparkConf
from pyspark.sql import SparkSession
from pyspark.sql.functions import col,sum,avg,max
import pyspark.sql.functions as F
from pyspark.sql.types import BooleanType
from pyspark.sql.functions import split, explode

conf = pyspark.SparkConf().setAppName('appName').setMaster('local')
#sc = pyspark.SparkContext(conf=conf)
sc = SparkContext.getOrCreate()
spark = SparkSession(sc)
csv_df = spark.read.option("header", "True").csv("./web.csv")
csv_df.printSchema()
```

```
root
 |-- IP: string (nullable = true)
 |-- Time: string (nullable = true)
 |-- URL: string (nullable = true)
 |-- Staus: string (nullable = true)

+-----+-----+
|Staus|count|
+-----+-----+
|  200|11330|
|  206|   52|
|  302| 3498|
|  404|  251|
|  304|  658|
+-----+-----+
```

[136]:
```
csv_df.registerTempTable("df")

# Print complete table
spark.sql("SELECT * from df").show()

# Only process URL and Status from table, discarding result code != 200
only_200_result = spark.sql("SELECT URL, Staus from df WHERE Staus!=200")

# Print result of query
#spark.sql("SELECT URL, Staus from df WHERE Staus!=200").show()

# Split command from path

split_URL = pyspark.sql.functions.split(only_200_result['URL'], ' ')
url_df = only_200_result.withColumn('URL_COMMAND', split_URL.getItem(1))

# Register new table to spark sql query
url_df.registerTempTable("URL_Table")

# Obtain list of 10 paths and counts in desc order.
top_ten_200_result = spark.sql("SELECT Staus, URL_COMMAND from URL_Table")\
                                .groupBy("URL_COMMAND")\
                                .count().\
                                orderBy(col("count").desc())\
                                .take(10)

# Print result of query
spark.sql("SELECT Staus, URL_COMMAND from URL_Table")\
                                .groupBy("URL_COMMAND")\
                                .count().\
                                orderBy(col("count").desc()).show(10)

print("*********************************************************")
print("Top 10 visited paths:")
print("*********************************************************")

# Print elements of the list
for i in top_ten_200_result:
    print(i)
```

```
+-----------+--------------------+--------------------+-----+
|         IP|                Time|                 URL|Staus|
+-----------+--------------------+--------------------+-----+
|10.128.2.1|[29/Nov/2017:06:5…|GET /login.php HT…|  200|
|10.128.2.1|[29/Nov/2017:06:5…|POST /process.php…|  302|
|10.128.2.1|[29/Nov/2017:06:5…|GET /home.php HTT…|  200|
|10.131.2.1|[29/Nov/2017:06:5…|GET /js/vendor/mo…|  200|
|10.130.2.1|[29/Nov/2017:06:5…|GET /bootstrap-3…|  200|
|10.130.2.1|[29/Nov/2017:06:5…|GET /profile.php?…|  200|
|10.128.2.1|[29/Nov/2017:06:5…|GET /js/jquery.mi…|  200|
|10.131.2.1|[29/Nov/2017:06:5…|GET /js/chart.min…|  200|
|10.131.2.1|[29/Nov/2017:06:5…|GET /edit.php?nam…|  200|
|10.131.2.1|[29/Nov/2017:06:5…|GET /logout.php H…|  302|
|10.131.2.1|[29/Nov/2017:06:5…|GET /login.php HT…|  200|
|10.130.2.1|[29/Nov/2017:07:0…|GET /login.php HT…|  200|
|10.130.2.1|[29/Nov/2017:07:0…|GET /login.php HT…|  200|
|10.130.2.1|[29/Nov/2017:13:3…|      GET / HTTP/1.1|  302|
|10.130.2.1|[29/Nov/2017:13:3…|GET /login.php HT…|  200|
|10.129.2.1|[29/Nov/2017:13:3…|POST /process.php…|  302|
|10.131.0.1|[29/Nov/2017:13:3…|GET /home.php HTT…|  200|
|10.131.0.1|[29/Nov/2017:13:3…|GET /contestprobl…|  200|
|10.130.2.1|[29/Nov/2017:13:3…|      GET / HTTP/1.1|  302|
|10.131.2.1|[29/Nov/2017:13:3…|GET /login.php HT…|  200|
+----------+--------------------+--------------------+-----+
only showing top 20 rows

+--------------------+-----+
|         URL_COMMAND|count|
+--------------------+-----+
|           /home.php| 2167|
|                   /|  741|
|        /process.php|  317|
|         /robots.txt|  224|
|         /action.php|   83|
|/contestproblem.p…|   74|
|/js/vendor/jquery…|   73|
|/css/bootstrap.mi…|   72|
|/js/vendor/modern…|   72|
|/css/font-awesome…|   68|
+--------------------+-----+
only showing top 10 rows

*******************************************************
Top 10 visited paths:
*******************************************************
Row(URL_COMMAND='/home.php', count=2167)
Row(URL_COMMAND='/', count=741)
Row(URL_COMMAND='/process.php', count=317)
Row(URL_COMMAND='/robots.txt', count=224)
Row(URL_COMMAND='/action.php', count=83)
Row(URL_COMMAND='/contestproblem.php?name=RUET%20OJ%20Server%20Testing%20Contest
', count=74)
Row(URL_COMMAND='/js/vendor/jquery-1.12.0.min.js', count=73)
Row(URL_COMMAND='/css/bootstrap.min.css', count=72)
Row(URL_COMMAND='/js/vendor/modernizr-2.8.3.min.js', count=72)
Row(URL_COMMAND='/css/main.css', count=68)
```

[64]:
```
ip_df = spark.sql("SELECT DISTINCT IP from df")
import ipaddress

@F.udf(returnType=BooleanType())
def is_ip(param_ip):
    try:
        ipaddress.ip_address(param_ip)
        return True
    except:
        return False

print("********************************************************")
print("Unique hosts:")
print("********************************************************")
ip_df.where(is_ip(col("IP"))==True).show()
```

```
********************************************************
Unique hosts:
********************************************************
+-----------+
|         IP|
+-----------+
|10.131.2.1|
|10.128.2.1|
|10.130.2.1|
|10.131.0.1|
|10.129.2.1|
+-----------+
```

[105]:
```
hosts_time= spark.sql("SELECT IP, Time from df")
# Split command from path
from pyspark.sql.functions import regexp_replace

# Split Time column by /
host_time_splitted = pyspark.sql.functions.split(hosts_time['Time'], '/')

# Get Day and Month columns
host_days = hosts_time.withColumn('Day', host_time_splitted.getItem(0)).
 ↪withColumn('Month', host_time_splitted.getItem(1))

# Replace [ character by nothing
host_days = host_days.withColumn('Day', regexp_replace(col('Day'), "\\[", ""))

# Drop time column (not useful)
```

```python
host_days = host_days.drop('Time')

# Drop invalid IP hosts
host_days = host_days.where(is_ip(col("IP"))==True)

# Group by Day and Month for different hosts
host_days = host_days.groupBy("Day", "Month").agg(F.countDistinct('IP'))

# Order by months
host_days = host_days.orderBy("Month")


print("*******************************************************")
print("Number of access by different hosts per day/month:")
print("*******************************************************")
# Show result from query
host_days.show()
```

```
*******************************************************
Number of access by different hosts per day/month:
*******************************************************
+---+-----+---------+
|Day|Month|count(IP)|
+---+-----+---------+
| 02|  Dec|        5|
| 03|  Dec|        5|
| 20|  Dec|        3|
| 23|  Dec|        3|
| 22|  Dec|        3|
| 21|  Dec|        3|
| 01|  Dec|        5|
| 15|  Dec|        3|
| 12|  Dec|        3|
| 18|  Dec|        3|
| 17|  Dec|        3|
| 14|  Dec|        3|
| 16|  Dec|        3|
| 19|  Dec|        3|
| 13|  Dec|        3|
| 27|  Feb|        3|
| 16|  Feb|        3|
| 21|  Feb|        3|
| 18|  Feb|        3|
| 25|  Feb|        3|
+---+-----+---------+
only showing top 20 rows
```

[108]:
```
responses_304 = spark.sql("SELECT count(*) as 304_Responses_Count from df WHERE␣
 ↪Staus=304")
responses_304.show()

print("*****************************************************")
print("Number of 304 responses:")
print("*****************************************************")
```

```
+-------------------+
|304_Responses_Count|
+-------------------+
|                658|
+-------------------+
```

[135]:
```
# Print complete table
spark.sql("SELECT * from df").show()

# Only process URL and Status from table, discarding result code != 200
only_304_result = spark.sql("SELECT URL, Staus from df WHERE Staus==304")

# Print result of query
#spark.sql("SELECT URL, Staus from df WHERE Staus==304").show()

# Split command from path

split_URL = pyspark.sql.functions.split(only_304_result['URL'], ' ')
url_df = only_304_result.withColumn('URL_COMMAND', split_URL.getItem(1))

# Register new table to spark sql query
url_df.registerTempTable("URL_Table")

# Obtain list of 10 paths and counts in desc order.
top_five_304_result = spark.sql("SELECT Staus, URL_COMMAND from URL_Table")\
                                .groupBy("URL_COMMAND")\
                                .count().\
                                orderBy(col("count").desc())\
                                .take(10)

print("*****************************************************")
print("Top five requested paths :")
print("*****************************************************")
# Print result of query
spark.sql("SELECT Staus, URL_COMMAND from URL_Table")\
                                .groupBy("URL_COMMAND")\
                                .count().\
                                orderBy(col("count").desc()).show(5)
```

```
+----------+--------------------+--------------------+-----+
|        IP|                Time|                 URL|Staus|
+----------+--------------------+--------------------+-----+
|10.128.2.1|[29/Nov/2017:06:5…|GET /login.php HT…|  200|
|10.128.2.1|[29/Nov/2017:06:5…|POST /process.php…|  302|
|10.128.2.1|[29/Nov/2017:06:5…|GET /home.php HTT…|  200|
|10.131.2.1|[29/Nov/2017:06:5…|GET /js/vendor/mo…|  200|
|10.130.2.1|[29/Nov/2017:06:5…|GET /bootstrap-3…|  200|
|10.130.2.1|[29/Nov/2017:06:5…|GET /profile.php?…|  200|
|10.128.2.1|[29/Nov/2017:06:5…|GET /js/jquery.mi…|  200|
|10.131.2.1|[29/Nov/2017:06:5…|GET /js/chart.min…|  200|
|10.131.2.1|[29/Nov/2017:06:5…|GET /edit.php?nam…|  200|
|10.131.2.1|[29/Nov/2017:06:5…|GET /logout.php H…|  302|
|10.131.2.1|[29/Nov/2017:06:5…|GET /login.php HT…|  200|
|10.130.2.1|[29/Nov/2017:07:0…|GET /login.php HT…|  200|
|10.130.2.1|[29/Nov/2017:07:0…|GET /login.php HT…|  200|
|10.130.2.1|[29/Nov/2017:13:3…|      GET / HTTP/1.1|  302|
|10.130.2.1|[29/Nov/2017:13:3…|GET /login.php HT…|  200|
|10.129.2.1|[29/Nov/2017:13:3…|POST /process.php…|  302|
|10.131.0.1|[29/Nov/2017:13:3…|GET /home.php HTT…|  200|
|10.131.0.1|[29/Nov/2017:13:3…|GET /contestprobl…|  200|
|10.130.2.1|[29/Nov/2017:13:3…|      GET / HTTP/1.1|  302|
|10.131.2.1|[29/Nov/2017:13:3…|GET /login.php HT…|  200|
+----------+--------------------+--------------------+-----+
only showing top 20 rows

********************************************************
Top five requested paths :
********************************************************
+--------------------+-----+
|         URL_COMMAND|count|
+--------------------+-----+
|/js/vendor/modern…|   71|
|/js/vendor/jquery…|   69|
|       /css/main.css|   68|
|/css/font-awesome…|   68|
|/css/bootstrap.mi…|   68|
+--------------------+-----+
only showing top 5 rows
```

[134]:
```python
# Print complete table
spark.sql("SELECT * from df").show()

# Only process URL and Status from table, discarding result code == 304
only_304_result = spark.sql("SELECT IP, Staus from df WHERE Staus==304")
```

```
# Print result of query
#spark.sql("SELECT IP, Staus from df WHERE Staus==304").show()


# Register new table to spark sql query
only_304_result.registerTempTable("IP_CODE_Table")

# Obtain list of 10 paths and counts in desc order.
top_five_304_result = spark.sql("SELECT * from IP_CODE_Table")\
                                .groupBy("IP")\
                                .count().\
                                orderBy(col("count").desc())\
                                .take(5)

print("********************************************************")
print("Top five hosts with 304 response :")
print("********************************************************")
# Print result of query
spark.sql("SELECT * from IP_CODE_Table")\
                                .groupBy("IP")\
                                .count().\
                                orderBy(col("count").desc()).show(5)
```

```
+----------+--------------------+--------------------+-----+
|        IP|                Time|                 URL|Staus|
+----------+--------------------+--------------------+-----+
|10.128.2.1|[29/Nov/2017:06:5...|GET /login.php HT...|  200|
|10.128.2.1|[29/Nov/2017:06:5...|POST /process.php...|  302|
|10.128.2.1|[29/Nov/2017:06:5...|GET /home.php HTT...|  200|
|10.131.2.1|[29/Nov/2017:06:5...|GET /js/vendor/mo...|  200|
|10.130.2.1|[29/Nov/2017:06:5...|GET /bootstrap-3...|  200|
|10.130.2.1|[29/Nov/2017:06:5...|GET /profile.php?...|  200|
|10.128.2.1|[29/Nov/2017:06:5...|GET /js/jquery.mi...|  200|
|10.131.2.1|[29/Nov/2017:06:5...|GET /js/chart.min...|  200|
|10.131.2.1|[29/Nov/2017:06:5...|GET /edit.php?nam...|  200|
|10.131.2.1|[29/Nov/2017:06:5...|GET /logout.php H...|  302|
|10.131.2.1|[29/Nov/2017:06:5...|GET /login.php HT...|  200|
|10.130.2.1|[29/Nov/2017:07:0...|GET /login.php HT...|  200|
|10.130.2.1|[29/Nov/2017:07:0...|GET /login.php HT...|  200|
|10.130.2.1|[29/Nov/2017:13:3...|      GET / HTTP/1.1|  302|
|10.130.2.1|[29/Nov/2017:13:3...|GET /login.php HT...|  200|
|10.129.2.1|[29/Nov/2017:13:3...|POST /process.php...|  302|
|10.131.0.1|[29/Nov/2017:13:3...|GET /home.php HTT...|  200|
|10.131.0.1|[29/Nov/2017:13:3...|GET /contestprobl...|  200|
|10.130.2.1|[29/Nov/2017:13:3...|      GET / HTTP/1.1|  302|
|10.131.2.1|[29/Nov/2017:13:3...|GET /login.php HT...|  200|
+----------+--------------------+--------------------+-----+
only showing top 20 rows

********************************************************
Top five hosts with 304 response :
********************************************************
+----------+-----+
|        IP|count|
+----------+-----+
|10.128.2.1|  148|
|10.130.2.1|  147|
|10.131.0.1|  131|
|10.131.2.1|  120|
|10.129.2.1|  112|
+----------+-----+
```

[133]:
```python
# Print complete table
spark.sql("SELECT * from df").show()

# Only process URL and Status from table, discarding result code != 200
only_two_codes_result = spark.sql("SELECT Time, Staus from df WHERE Staus==206␣
 ↪OR Staus==404")

# Print result of query
#spark.sql("SELECT Time, Staus from df WHERE Staus==206 OR Staus==404").show()

# Split Time column by /
splitted_time = pyspark.sql.functions.split(only_two_codes_result['Time'], '/')

# Get Day and Month columns
only_two_codes_result = only_two_codes_result.withColumn('Day', splitted_time.
 ↪getItem(0)).withColumn('Month', splitted_time.getItem(1))

only_two_codes_result = only_two_codes_result.drop('Time')

# Replace [ character by nothing
only_two_codes_result = only_two_codes_result.withColumn('Day',␣
 ↪regexp_replace(col('Day'), "\\[", ""))

# Group by Day and Month for different hosts
only_two_codes_result = only_two_codes_result.groupBy("Day", "Month", "Staus").
 ↪count()

only_two_codes_result = only_two_codes_result.orderBy("Month")

print("********************************************************")
print("Day/Month with 206/404 responses :")
print("*******************************************************")
only_two_codes_result.show(1000)
```

```
+----------+--------------------+--------------------+-----+
|        IP|                Time|                 URL|Staus|
+----------+--------------------+--------------------+-----+
|10.128.2.1|[29/Nov/2017:06:5...|GET /login.php HT...|  200|
|10.128.2.1|[29/Nov/2017:06:5...|POST /process.php...|  302|
|10.128.2.1|[29/Nov/2017:06:5...|GET /home.php HTT...|  200|
|10.131.2.1|[29/Nov/2017:06:5...|GET /js/vendor/mo...|  200|
|10.130.2.1|[29/Nov/2017:06:5...|GET /bootstrap-3...|  200|
|10.130.2.1|[29/Nov/2017:06:5...|GET /profile.php?...|  200|
|10.128.2.1|[29/Nov/2017:06:5...|GET /js/jquery.mi...|  200|
|10.131.2.1|[29/Nov/2017:06:5...|GET /js/chart.min...|  200|
|10.131.2.1|[29/Nov/2017:06:5...|GET /edit.php?nam...|  200|
|10.131.2.1|[29/Nov/2017:06:5...|GET /logout.php H...|  302|
|10.131.2.1|[29/Nov/2017:06:5...|GET /login.php HT...|  200|
|10.130.2.1|[29/Nov/2017:07:0...|GET /login.php HT...|  200|
|10.130.2.1|[29/Nov/2017:07:0...|GET /login.php HT...|  200|
|10.130.2.1|[29/Nov/2017:13:3...|      GET / HTTP/1.1|  302|
|10.130.2.1|[29/Nov/2017:13:3...|GET /login.php HT...|  200|
|10.129.2.1|[29/Nov/2017:13:3...|POST /process.php...|  302|
|10.131.0.1|[29/Nov/2017:13:3...|GET /home.php HTT...|  200|
|10.131.0.1|[29/Nov/2017:13:3...|GET /contestprobl...|  200|
|10.130.2.1|[29/Nov/2017:13:3...|      GET / HTTP/1.1|  302|
|10.131.2.1|[29/Nov/2017:13:3...|GET /login.php HT...|  200|
+----------+--------------------+--------------------+-----+
only showing top 20 rows

*******************************************************
Day/Month with 206/404 responses :
*******************************************************
+---+-----+-----+-----+
|Day|Month|Staus|count|
+---+-----+-----+-----+
| 03|  Dec|  206|    6|
| 01|  Dec|  404|    5|
| 13|  Dec|  404|    5|
| 21|  Dec|  404|   23|
| 12|  Dec|  206|    1|
| 16|  Dec|  404|    4|
| 23|  Dec|  404|    1|
| 15|  Dec|  404|    2|
| 02|  Dec|  404|    3|
| 17|  Dec|  404|    3|
| 12|  Dec|  404|    2|
| 19|  Dec|  404|    3|
| 21|  Dec|  206|    1|
| 18|  Dec|  404|    2|
| 20|  Dec|  404|    7|
| 14|  Dec|  404|    4|
| 03|  Dec|  404|    3|
| 17|  Feb|  404|    4|
| 24|  Feb|  404|    2|
| 23|  Feb|  206|    2|
| 15|  Feb|  404|    1|
| 21|  Feb|  404|    3|
| 22|  Feb|  206|    2|
| 26|  Feb|  404|    2|
| 27|  Feb|  404|    2|
| 20|  Feb|  404|    3|
| 25|  Feb|  404|    3|
| 23|  Feb|  404|    2|
| 16|  Feb|  404|    3|
| 18|  Feb|  404|    1|
| 19|  Feb|  404|    2|
| 28|  Feb|  404|    3|
| 22|  Feb|  404|    3|
| 18|  Jan|  404|    1|
| 16|  Jan|  404|    1|
| 17|  Jan|  404|    4|
| 16|  Jan|  206|    2|
| 29|  Jan|  404|    1|
| 02|  Mar|  404|    2|
| 01|  Mar|  404|    2|
| 16|  Nov|  404|   10|
| 23|  Nov|  206|    1|
| 10|  Nov|  404|    6|
| 22|  Nov|  404|    3|
| 23|  Nov|  404|    2|
| 25|  Nov|  404|    7|
| 17|  Nov|  206|    1|
| 14|  Nov|  404|   18|
| 18|  Nov|  404|    4|
| 11|  Nov|  404|   12|
| 21|  Nov|  404|    2|
| 08|  Nov|  404|    8|
| 13|  Nov|  404|   10|
| 29|  Nov|  404|    1|
| 12|  Nov|  404|    6|
| 26|  Nov|  404|    6|
| 17|  Nov|  404|    5|
| 09|  Nov|  404|   10|
| 20|  Nov|  404|    1|
| 24|  Nov|  404|    3|
| 19|  Nov|  404|    5|
| 15|  Nov|  404|    8|
| 30|  Nov|  404|   12|
| 29|  Nov|  206|    2|
| 30|  Nov|  206|   34|
+---+-----+-----+-----+
```

[142]:

```
# Print complete table
spark.sql("SELECT * from df").show()

# Only process Time and Status from table with result code == 206 or 404
only_two_codes_by_hour_result = spark.sql("SELECT Time, Staus from df WHERE␣
 ↪Staus==206 OR Staus==404")

# Print result of query
#spark.sql("SELECT Time, Staus from df WHERE Staus==206 OR Staus==404").show()

# Split Time column by /
splitted_time = pyspark.sql.functions.
 ↪split(only_two_codes_by_hour_result['Time'], ':')

# Get Hour from Time Column
only_two_codes_by_hour_result = only_two_codes_by_hour_result.
 ↪withColumn('Hour', splitted_time.getItem(1))

# Drop Time Column (not useful)
only_two_codes_by_hour_result = only_two_codes_by_hour_result.drop('Time')

# Group by Hour and Status
only_two_codes_by_hour_result = only_two_codes_by_hour_result.groupBy("Hour",␣
 ↪"Staus").count()

# Order by Hour
only_two_codes_by_hour_result = only_two_codes_by_hour_result.orderBy("Hour")

print("********************************************************")
print("Hours with 206/404 responses :")
print("********************************************************")
only_two_codes_by_hour_result.show(1000)
```

```
+----------+--------------------+--------------------+-----+
|        IP|                Time|                 URL|Staus|
+----------+--------------------+--------------------+-----+
|10.128.2.1|[29/Nov/2017:06:5…|GET /login.php HT…|  200|
|10.128.2.1|[29/Nov/2017:06:5…|POST /process.php…|  302|
|10.128.2.1|[29/Nov/2017:06:5…|GET /home.php HTT…|  200|
|10.131.2.1|[29/Nov/2017:06:5…|GET /js/vendor/mo…|  200|
|10.130.2.1|[29/Nov/2017:06:5…|GET /bootstrap-3…|  200|
|10.130.2.1|[29/Nov/2017:06:5…|GET /profile.php?…|  200|
|10.128.2.1|[29/Nov/2017:06:5…|GET /js/jquery.mi…|  200|
|10.131.2.1|[29/Nov/2017:06:5…|GET /js/chart.min…|  200|
|10.131.2.1|[29/Nov/2017:06:5…|GET /edit.php?nam…|  200|
|10.131.2.1|[29/Nov/2017:06:5…|GET /logout.php H…|  302|
|10.131.2.1|[29/Nov/2017:06:5…|GET /login.php HT…|  200|
|10.130.2.1|[29/Nov/2017:07:0…|GET /login.php HT…|  200|
|10.130.2.1|[29/Nov/2017:07:0…|GET /login.php HT…|  200|
|10.130.2.1|[29/Nov/2017:13:3…|      GET / HTTP/1.1|  302|
|10.130.2.1|[29/Nov/2017:13:3…|GET /login.php HT…|  200|
|10.129.2.1|[29/Nov/2017:13:3…|POST /process.php…|  302|
|10.131.0.1|[29/Nov/2017:13:3…|GET /home.php HTT…|  200|
|10.131.0.1|[29/Nov/2017:13:3…|GET /contestprobl…|  200|
|10.130.2.1|[29/Nov/2017:13:3…|      GET / HTTP/1.1|  302|
|10.131.2.1|[29/Nov/2017:13:3…|GET /login.php HT…|  200|
+----------+--------------------+--------------------+-----+
only showing top 20 rows

*******************************************************
Hours with 206/404 responses :
*******************************************************
+----+-----+-----+
|Hour|Staus|count|
+----+-----+-----+
|  00|  404|   12|
|  01|  404|    4|
|  02|  404|    5|
|  03|  404|    3|
|  04|  404|    5|
|  05|  404|    4|
|  06|  404|   18|
|  07|  404|    9|
|  07|  206|    2|
|  08|  404|   14|
|  09|  404|    6|
|  10|  404|   10|
|  10|  206|    2|
|  11|  404|   11|
|  12|  206|    6|
|  12|  404|    8|
|  13|  206|   16|
|  13|  404|    9|
|  14|  404|    9|
|  14|  206|    4|
|  15|  404|   32|
|  15|  206|   13|
|  16| 206|    2|
|  16| 404|   19|
|  17| 404|    5|
|  18| 404|    9|
|  18| 206|    5|
|  19| 404|   14|
|  20| 404|   15|
|  21| 404|    9|
|  22| 404|    7|
|  23| 404|   14|
|  23| 206|    2|
+----+-----+-----+
```

[ ]:

```

```