

Big Data Analytics: HW#4

108590450 羅傑 資工三

108590452 林峻霆 資工三

Programming Exercise in MapReduce

Goal: A MapReduce program for analyzing the check-in records in social networks

Input: Check-in records in social networking site *Brightkite*

- Time and location information of check-ins made by users
- Friendship network of Brightkite users

Output: results of analysis (to be detailed later)

Detailed analysis including the following subtasks:

```
In [1]: # user_id, user_id
edges_path = "Brightkite_edges.txt"
# user_id, checkin_time, latitude, longitude, location_id
totalCheckins_path = "Brightkite_totalCheckins.txt"

In [2]: from pyspark import SparkContext
import time
start_time = time.time()
print("--- %s seconds ---" % (time.time() - start_time))
sc = SparkContext("local", "HW4_1")

# 輸入檔案
text_file = sc.textFile(totalCheckins_path)

# word count
counts = text_file.map( lambda line: (line.lower().split("\t")[4],1) ) \
    .reduceByKey( lambda a, b: a + b ) \
    .sortBy( lambda x: x[1], False )
output = counts.collect()

for (location, count) in output:
    print( f'({location}, {count})' )
# Stopping Spark Context
sc.stop()
print("--- %s seconds ---" % (time.time() - start_time))

((f139822171111de4e5003048c0801e, 1)
(edd1bed6a22411ddbed663011b1bd08d, 1)
(3d7d14b25e8ac8b547b025c5c095abab, 1)
(d96221c0a22411dd8663b716ce2a00d1, 1)
(eedbf7eea22411dd8f6a23d05d0a1909, 1)
(298c07c1a33a748b0b04c259d60806a505403e58, 1)
(838eb604dfe8ede02bd9a5843454c3c, 1)
(99f4245479ea11de84f3003048c10834, 1)
(7008cf44678811de9323003048c10834, 1)
(d08263f86f4b11deaa6003048c10834, 1)
(9f7c05905e7911de9102003048c10834, 1)
(3748d39ca2f711dd85eb003048c0801e, 1)
(d4e293a6aef142f8018f2cf33d5a43e, 1)
(3b3eac1a08572ef2d9cd4594b85582b, 1)
(a1c0f9a853f110d9072003048c0801e, 1)
(6c416d83cb311de947a003048c0801e, 1)
(b448102831aa11de9e7b003048c10834, 1)

--- 117.9945616722107 seconds ---
```

List the top checked-in *locations* (most popular)

```
In [3]: from pyspark import SparkContext
import time
start_time = time.time()
print("--- %s seconds ---" % (time.time() - start_time))
sc = SparkContext("local", "HW4_2")

# 輸入檔案
text_file = sc.textFile(totalCheckins_path)

# word count
counts = text_file.map( lambda line: (line.lower().split("\t")[0],1) ) \
    .reduceByKey( lambda a, b: a + b ) \
    .sortBy( lambda x: x[1], False )
output = counts.collect()

for (location, count) in output:
    print( f'({location}, {count})' )
# Stopping Spark Context
sc.stop()
print("--- %s seconds ---" % (time.time() - start_time))

(56958, 1)
(56994, 1)
(56998, 1)
(57005, 1)
(57054, 1)
(57285, 1)
(57319, 1)
(57324, 1)
(57414, 1)
(57432, 1)
(57469, 1)
(57670, 1)
(57806, 1)
(57850, 1)
(57942, 1)
(57987, 1)
(58019, 1)
(58054, 1)

--- 16.8202223777771 seconds ---
```

List the top checked-in **users**

List the most popular **time** for check-ins (where time is divided into intervals by hours, for example, 7:00-8:00 or 18:00-19:00)

```
In [4]: def intervalByHour(line):
        try:
            import datetime
            string = line.split("\t")[1]
            element = datetime.datetime.strptime(string, "%Y-%m-%dT%H:%M:%SZ")

            hourStart = str(element.hour).rjust(2, "0") + ":00"
            hourEnd = ("00" if element.hour+1 == 24 else str(element.hour+1).rjust(2, "0")) + ":00"
            return (f"{hourStart}-{hourEnd}", 1)
        except:
            return ("Error", 1)
        # print(f"{element.hour}/{element.minute}/{element.second}")
        # print(line)
        # print(f"{element.hour}/{element.minute}/{element.second}")

from pyspark import SparkContext
import time
start_time = time.time()
print("--- %s seconds ---" % (time.time() - start_time))
sc = SparkContext("local", "HW4_3")

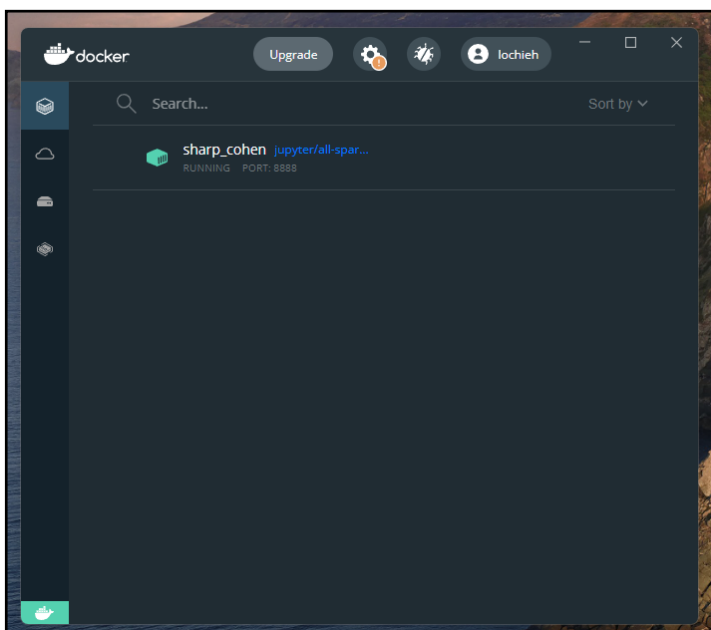
# 載入檔案
text_file = sc.textFile(totalCheckins_path)




# word count
counts = text_file.map(intervalByHour) \
    .reduceByKey(lambda a, b: a + b) \
    .sortBy(lambda x: x[1], False)
output = counts.collect()

for (interval, count) in output:
    print(f'({interval}, {count})')
# Stopping Spark Context
sc.stop()
print("--- %s seconds ---" % (time.time() - start_time))

--- 1.8596649169921875e-05 seconds ---

(00:00-01:00, 249610)
(18:00-19:00, 249547)
(23:00-00:00, 246815)
(19:00-20:00, 244507)
(01:00-02:00, 244008)
(17:00-18:00, 241841)
(22:00-23:00, 240323)
(20:00-21:00, 238403)
(21:00-22:00, 235774)
(02:00-03:00, 233628)
(16:00-17:00, 226600)
(03:00-04:00, 224873)
(15:00-16:00, 211702)
(04:00-05:00, 203360)
(14:00-15:00, 196438)
(05:00-06:00, 174204)
(13:00-14:00, 170380)
(06:00-07:00, 150076)
(12:00-13:00, 142481)
(07:00-08:00, 135464)
(08:00-09:00, 126108)
(11:00-12:00, 124127)
(09:00-10:00, 120083)
(10:00-11:00, 116923)
(Error, 6)
--- 45.372206926345825 seconds ---
```



	practical_kepler	jupyter/pyspark...	RUNNING
	interesting_grothendieck	jupyter/all-spar...	RUNNING PORT: 8888
	databasesystemproject-backend		RUNNING

Your cluster environment setup.

Docker Image Properties

Name	Tag	IMAGE ID	CREATED	SIZE
jupyter/all-spark-notebook	Latest	0245f58cc2c3	about 1 month ago	4.26 GB
jupyter/all-spark-notebook	Latest	c7a3ce5cab39	3 days ago	3.44 GB

How many PCs (or VMs), hardware spec (CPU cores, memory, storage), network setup, ...

重新命名此電腦

裝置規格

複製

裝置名稱

DESKTOP-7PC9N7G

處理器

11th Gen Intel(R) Core(TM) i5-11400F @ 2.60GHz 2.59 GHz

已安裝記憶體(RAM)

16.0 GB (15.9 GB 可用)

裝置識別碼

產品識別碼

系統類型

64 位元作業系統, x64 型處理器

手寫筆與觸控

此顯示器不提供手寫筆或觸控式輸入功能

相關連結

網路或工作群組

系統保護

進階系統設定

Windows 規格

複製

版本

Windows 11 專業工作站版 Insider Preview

版本

Dev

安裝於

2021/12/18

OS 組建

22523.1000

體驗

Windows Feature Experience Pack 1000.22523.1000.0

Microsoft 服務合約

Microsoft 軟體授權條款

MacBook Pro

硬體

ATA

Apple Pay

FireWire

NVMeExpress

PCI

SAS

SATA

SPI

Thunderbolt/USB4

USB

乙太網路

儲存裝置

光纖通道

印表機

平行 SCSI

控制器

燒錄光碟

相機

藍牙

硬體概覽：

機型名稱：

MacBook Pro

機型識別碼：

MacBookPro16,3

處理器名稱：

四核心 Intel Core i5

處理器速度：

1.4 GHz

處理器數目：

1

總核心數目：

4

L2 快取記憶體 (每個核心)：

256 KB

L3 快取記憶體：

6 MB

超執行緒技術：

已啟用

記憶體：

8 GB

系統韌體版本：

1715.40.15.0.0 (iBridge: 19.16.10548.0.0.0)

OS 裝載程式版本：

540.40.4~45

序號 (系統)：

硬體 UUID：

佈建 UUID：

啟用鎖定狀態：

已啟用

Your **source code**
In folder name **HW4.ipynb**.

Documentation on how to compile, install, or configure the environment
Windows :

安裝

您現在可以在系統管理員 PowerShell 或 Windows 命令提示字元中輸入此命令，然後重新開機電腦，安裝執行 Windows 子系統 Linux 版 (WSL) 所需的所有專案。

PowerShell

複製

```
wsl --install
```

此命令會啟用必要的選擇性元件、下載最新的 Linux 核心、將 WSL 2 設定為預設值，並根據預設為您安裝 Linux 發行版本 (Ubuntu，請參閱下方的變更此)。

當您第一次啟動新安裝的 Linux 發行版本時，主控台視窗將會開啟，並要求您等候檔案解除壓縮並儲存在您的電腦上。未來的所有啟動時間都應該會低於一秒。

在 WSL 2 上開始使用 Docker 遠端容器

發行項 • 2021/11/28

此頁面有所助益嗎？


本逐步指南將協助您開始使用遠端容器進行開發，方法是使用 WSL 2 (Windows 子系統 Linux 版第2版) 來設定適用於 Windows 的 Docker Desktop。

適用於 Windows 的 Docker Desktop 提供開發環境，可供建立、交付和執行 docker 化應用程式。藉由啟用 WSL 2 型引擎，您可以在同一部電腦上執行 Docker Desktop 中的 Linux 和 Windows 容器。(的 docker Desktop 免費供個人使用 and 小型企業使用，如需 Pro、小組或商務定價的詳細資訊，請參閱[Docker 網站常見問題](#))。

<https://docs.microsoft.com/zh-tw/windows/wsl/tutorials/wsl-containers>

Docker in Windows :

1. 確認支援Hyper-V
2. Docker Hub 進行註冊：<https://hub.docker.com/>
3. 下載Docker



Docker Desktop for Windows

By [Docker](#)

The fastest and easiest way to get started with Docker on Windows

Edition Windows x86-64

Get Docker Desktop for Windows


Requires Microsoft Windows 10 Professional or Enterprise 64-bit, or Windows 10 Home 64-bit with WSL 2.

We updated the [Docker Subscription Service Agreement](#) on August 31, 2021. Please read the announcement and FAQs to learn more.

[Subscription Service Agreement](#) | [Data Processing Agreement](#) | [Data Privacy Policy](#)

[Download Docker Desktop](#)

4. Docker 架設PySpark



jupyter/all-spark-notebook ☆

By [jupyter](#) • Updated 3 days ago

Jupyter Notebook Python, Scala, R, Spark, Mesos Stack from <https://github.com/jupyter/docker-stacks>

Container

\$ docker pull jupyter/all-spark-notebook

\$ docker run -p 8888:8888 jupyter/all-spark-notebook:IMAGEID

```
sharp.cohen jupyter/all-spark-notebook:latest
RUNNING

[I 08:07:24.968 NotebookApp] http://6125167ae0f8:8888/?token=c287af2634a518714db84edadaccf8ffb823c4484cc8dccc
[I 08:07:24.968 NotebookApp] or http://127.0.0.1:8888/?token=c287af2634a518714db84edadaccf8ffb823c4484cc8dccc
[I 08:07:24.968 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 08:07:24.971 NotebookApp]

To access the notebook, open this file in a browser:
    file:///home/jovyan/.local/share/jupyter/runtime/nbserver-7-open.html
Or copy and paste one of these URLs:
    http://6125167ae0f8:8888/?token=c287af2634a518714db84edadaccf8ffb823c4484cc8dccc
    or http://127.0.0.1:8888/?token=c287af2634a518714db84edadaccf8ffb823c4484cc8dccc
[I 08:07:33.234 NotebookApp] 302 GET /?token=c287af2634a518714db84edadaccf8ffb823c4484cc8dccc (172.17.0.1) 0.750000ms
Exception in callback <TaskWrapper object at 0x7f5160ea4940> (<Future finis...53b'\r\n\r\n'>)
handle: <Handle <TaskWrapper object at 0x7f5160ea4940> (<Future finis...53b'\r\n\r\n'>)>
Traceback (most recent call last):
  File "/opt/conda/lib/python3.9/site-packages/tornado/ioloop.py:688" in run
    self._context.run(self._callback, *self._args)
RuntimeError: Cannot enter into task <Task pending name='Task-2' coro=<HTTPServerConnection._server_request_loop() running at /opt/conda/lib/python3.9/site-packages/tornado/httpclient.py:823> wait_for=Future finished result=b'GET /api/co...d53b'\r\n\r\n'> cb=[IOLoop.add_future.<locals>.<lambda>() at /opt/conda/lib/python3.9/site-packages/tornado/ioloop.py:688]> while another task <Task pending name='Task-42' coro=<MappingKernelManager.start_kernel() running at /opt/conda/lib/python3.9/site-packages/notebook/services/kernels/kernelmanager.py:176> cb=[IOLoop.add_future.<locals>.<lambda>() at /opt/conda/lib/python3.9/site-packages/tornado/ioloop.py:688]> is being executed.
[I 08:07:45.748 NotebookApp] Kernel started: ce42925e-4ff2-4c3a-8b40-e99f224effeb, name: python3
[I 08:09:46.657 NotebookApp] Saving file at /work/HW4.ipynb
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/usr/local/spark-3.2.0-bin-hadoop3.2/jars/spark-unsafe_2.12-3.2.0.jar) to constructor java.nio.DirectByteBuffer(long,int)
```

Open file in a browser :

Copy and paste one of these URLs.
Start jupyter.

jupyter

Quit

Logout

Files

Running

IPython Clusters

Select items to perform actions on them.

0

/ work

Name


Last Modified

File size

..

幾秒前


☐

 HW4.ipynb

Running 1 小時前

39.7 MB


☐

 Brightkite_edges.txt

1 天前

4.58 MB

☐

 Brightkite_totalChecksins.txt

1 天前

382 MB

Docker in Mac OS :

1. Docker Hub 進行註冊 : <https://hub.docker.com/>
2. 下載Docker

Install Docker Desktop on Mac

Estimated reading time: 7 minutes

Update to the Docker Desktop terms

Professional use of Docker Desktop in large organizations (more than 250 employees or more than \$10 million in annual revenue) requires users to have a paid Docker subscription. While the effective date of these terms is August 31, 2021, there is a grace period until January 31, 2022, for those that require a paid subscription. For more information, see the blog [Docker is Updating and Extending Our Product Subscriptions and the Docker Desktop License Agreement](#).


Welcome to Docker Desktop for Mac. This page contains information about Docker Desktop for Mac system requirements, download URLs, instructions to install and update Docker Desktop for Mac.

Download Docker Desktop for Mac

Mac with Intel chip

Mac with Apple chip

3. Docker架設PySpark



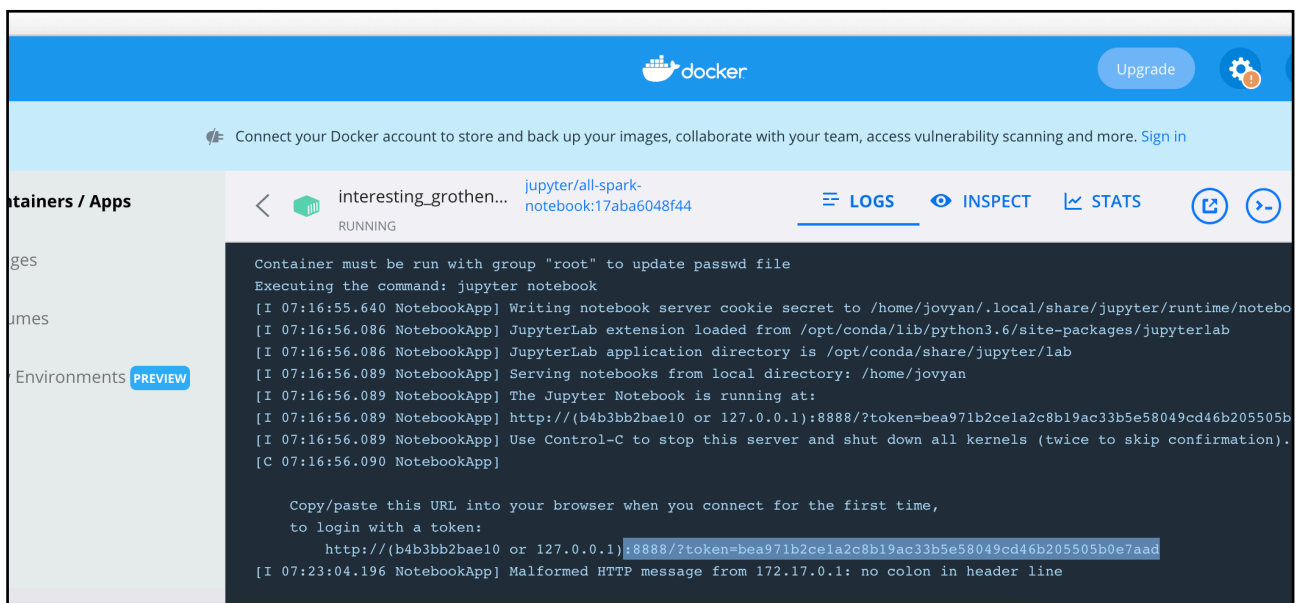
jupyter/all-spark-notebook ☆

By [jupyter](#) • Updated 3 days ago

Jupyter Notebook Python, Scala, R, Spark, Mesos Stack from <https://github.com/jupyter/docker-stacks>

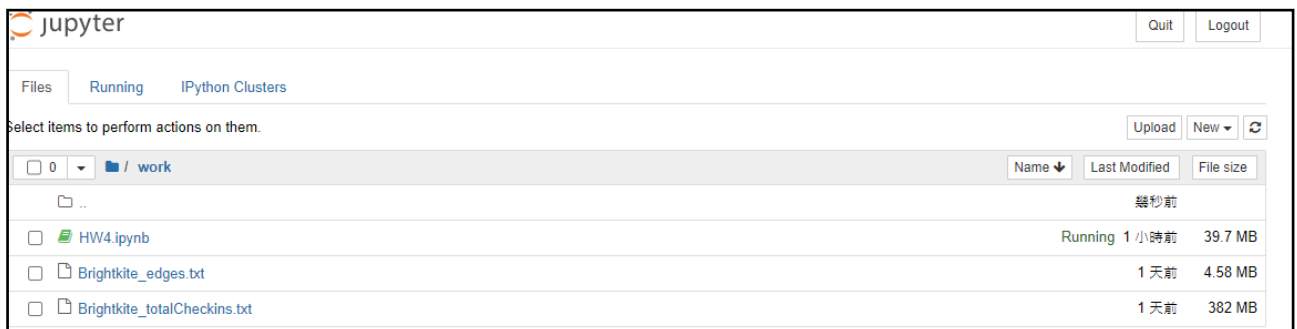
Container

```
$ docker pull jupyter/all-spark-notebook
$ docker run -p 8888:8888 jupyter/all-spark-notebook:IMAGEID
```

Open file in a browser :

Copy and paste one of these URLs.
Start jupyter.

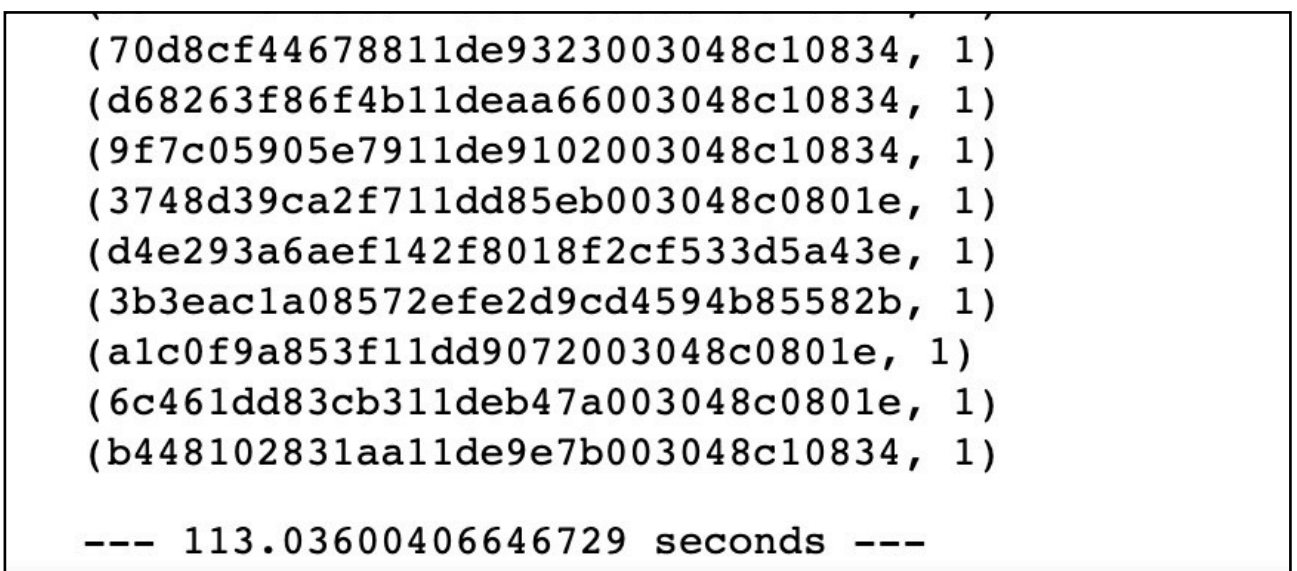


The **efficiency** of implemented algorithm will also be taken into account.

最一開始使用迴圈執行效率不佳，後來使用Spark map .reduceByKey() .sortBy() 和撰寫時間處理字串函數，將算法的時間複雜度降低。

兩台電腦算法執行時間比較

Macbook pro :



```
(57054, 1)
(57285, 1)
(57319, 1)
(57324, 1)
(57414, 1)
(57432, 1)
(57469, 1)
(57670, 1)
(57806, 1)
(57850, 1)
(57942, 1)
(57987, 1)
(58019, 1)
(58054, 1)
--- 19.108819246292114 seconds ---
```

```
(14:00-15:00, 196438)
(05:00-06:00, 174204)
(13:00-14:00, 170380)
(06:00-07:00, 150076)
(12:00-13:00, 142481)
(07:00-08:00, 135464)
(08:00-09:00, 126108)
(11:00-12:00, 124127)
(09:00-10:00, 120083)
(10:00-11:00, 116923)
(Error, 6)
--- 59.2446813583374 seconds ---
```

Windows :

```
(fa139822171311deb4e5003048c0801e, 1)
(edd1bed6a22411ddbed663011b1bd08d, 1)
(3d7d14b25e8ac8b547b825c5c095abab, 1)
(d96221c0a22411dd8663b716ce2a00d1, 1)
(eedbffeea22411dd8fea23d05d0a1969, 1)
(298c07c2a33a748bdb04c259d68086a505463e58, 1)
(838eb604dfe8e02bd9a5843454c3c, 1)
(99f4245479ea11de84f3003048c10834, 1)
(70d8cf44678811de9323003048c10834, 1)
(d68263f86f4b11deaa66003048c10834, 1)
(9f7c05905e7911de9102003048c10834, 1)
(3748d39ca2f711dd85eb003048c0801e, 1)
(d4e293a6aef142f8018f2cf533d5a43e, 1)
(3b3eac1a08572efe2d9cd4594b85582b, 1)
(a1c0f9a853f11dd9072003048c0801e, 1)
(6c461dd83cb311deb47a003048c0801e, 1)
(b448102831aa11de9e7b003048c10834, 1)
--- 117.9945616722107 seconds ---
```



```
(57285, 1)
(57319, 1)
(57324, 1)
(57414, 1)
(57432, 1)
(57469, 1)
(57670, 1)
(57806, 1)
(57850, 1)
(57942, 1)
(57987, 1)
(58019, 1)
(58054, 1)
--- 16.8202223777771 seconds ---
```

```
(04:00-05:00, 203366)
(14:00-15:00, 196438)
(05:00-06:00, 174204)
(13:00-14:00, 170380)
(06:00-07:00, 150076)
(12:00-13:00, 142481)
(07:00-08:00, 135464)
(08:00-09:00, 126108)
(11:00-12:00, 124127)
(09:00-10:00, 120083)
(10:00-11:00, 116923)
(Error, 6)
--- 45.372206926345825 seconds ---
```