L200921
23/04/24

Exercise 3.3.2

(a). $h_3(x) = 2x+4 \mod 5$ (b). $h_4(x) = 3x-1 \mod 5$

| Row | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $x+1 \mod 5$ | $3x+1 \mod 5$ | $2x+4 \mod 5$ | $3x-1 \mod 5$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 4 | -1 |
| 1 | 0 | 0 | 1 | 0 | 2 | 4 | 1 | 2 |
| 2 | 0 | 1 | 0 | 1 | 3 | 2 | 3 | 0 |
| 3 | 1 | 0 | 1 | 1 | 4 | 0 | 0 | 3 |
| 4 | 0 | 0 | 1 | 0 | 0 | 3 | 2 | 1 |

Signatures

| Hash | $S_1$ | $S_2$ | $S_3$ | $S_4$ |
|---|---|---|---|---|
| $h_3$ | 0 ~~4~~ ~~∞~~ | 3 ~~∞~~ | 0 ~~1~~ ~~∞~~ | 0 ~~3~~ ~~4~~ ~~∞~~ |
| $h_4$ | -1 ~~∞~~ | 0 ~~∞~~ | 1 ~~2~~ ~~∞~~ | -1 ~~∞~~ |

final Answer

| Hash | $S_1$ | $S_2$ | $S_3$ | $S_4$ |
|---|---|---|---|---|
| $h_3$ | 0 | 3 | 0 | 0 |
| $h_4$ | -1 | 0 | 1 | -1 |

Exercise 3.3.3

| Row | $2x+1 \mod 6$ | $3x+2 \mod 6$ | $5x+2 \mod 6$ |
|---|---|---|---|
| 0 | 1 | 2 | 2 |
| 1 | 3 | 5 | 1 |
| 2 | 5 | 2 | 0 |
| 3 | 1 | 5 | 5 |
| 4 | 3 | 2 | 4 |
| 5 | 5 | 5 | 3 |

2200921

23/04/24

Hash Signature

| hash | $S_1$ | $S_2$ | $S_3$ | $S_4$ |
|---|---|---|---|---|
| $h_1$ | 5 ~~∞~~ | 1 ~~∞~~ | 1 ~~∞~~ | 1 ~~∞~~ |
| $h_2$ | 2 ~~∞~~ | 2 ~~∞~~ | 2~~8~~ ~~∞~~ | 2 ~~∞~~ |
| $h_3$ | 0 ~~∞~~ | 1~~2~~ ~~∞~~ | 4 ~~∞~~ | 0~~2~~ ~~∞~~ |

final Answer

| hash | $S_1$ | $S_2$ | $S_3$ | $S_4$ |
|---|---|---|---|---|
| $h_1$ | 5 | 1 | 1 | 1 |
| $h_2$ | 2 | 2 | 2 | 2 |
| $h_3$ | 0 | 1 | 4 | 0 |

(b). True hash function is $h_3 = 5x+2 \bmod 6$ because it maps each distinct input value to a unique output value ~~and~~, distribution of hash values is uniform across the range and covers all possible output values within its range.

(c). Similarities.

| | 1-2 | 1-3 | 1-4 | 2-3 | 2-4 | 3-4 |
|---|---|---|---|---|---|---|
| col/col | 0 | 0 | 0.25 | 0 | 0.25 | 0.25 |
| Sig/sig | 0.333 | 0.333 | 0.667 | 0.667 | 0.6667 | 0.6667 |

$\Rightarrow$ Not all close to the true ones.

**Aisha Muhammad Nawaz L200921**
PySpark Class Activity 8A BSCS MMD
23rd April 2024

Instructions:

1. Write efficient Spark code for creating K-shingles given a huge document and K as input.
2. Write an efficient SPARK code for Minhashing (uses the logic of hash functions as shown in the uploaded slide). The map reduce code is given in slides.

In [1]:
```
# #Running on Colab
!pip install pyspark
!pip install -U -q PyDrive
!apt install openjdk-8-jdk-headless -qq
import os
os.environ['JAVA_HOME'] = '/usr/lib/jvm/java-8-openjdk-amd64'
```

```
Collecting pyspark
  Downloading pyspark-3.5.1.tar.gz (317.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 317.0/317.0 MB 3.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.5.1-py2.py3-none-any.whl size=317488491 sha256=46fb6e6cffc7c6d40c7dc511d904e7bcd1a55447f58431d4859af001a126bf37
  Stored in directory: /root/.cache/pip/wheels/80/1d/60/2c256ed38dddce2fdd93be545214a63e02fbd8d74fb0b7f3a6
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.5.1
The following additional packages will be installed:
  libxtst6 openjdk-8-jre-headless
Suggested packages:
  openjdk-8-demo openjdk-8-source libnss-mdns fonts-dejavu-extra fonts-nanum fonts-ipafont-gothic
  fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  libxtst6 openjdk-8-jdk-headless openjdk-8-jre-headless
0 upgraded, 3 newly installed, 0 to remove and 45 not upgraded.
Need to get 39.7 MB of archives.
After this operation, 144 MB of additional disk space will be used.
Selecting previously unselected package libxtst6:amd64.
(Reading database ... 121752 files and directories currently installed.)
Preparing to unpack .../libxtst6_2%3a1.2.3-1build4_amd64.deb ...
Unpacking libxtst6:amd64 (2:1.2.3-1build4) ...
Selecting previously unselected package openjdk-8-jre-headless:amd64.
Preparing to unpack .../openjdk-8-jre-headless_8u402-ga-2ubuntu1~22.04_amd64.deb ...
Unpacking openjdk-8-jre-headless:amd64 (8u402-ga-2ubuntu1~22.04) ...
Selecting previously unselected package openjdk-8-jdk-headless:amd64.
Preparing to unpack .../openjdk-8-jdk-headless_8u402-ga-2ubuntu1~22.04_amd64.deb ...
Unpacking openjdk-8-jdk-headless:amd64 (8u402-ga-2ubuntu1~22.04) ...
Setting up libxtst6:amd64 (2:1.2.3-1build4) ...
Setting up openjdk-8-jre-headless:amd64 (8u402-ga-2ubuntu1~22.04) ...
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/orbd to provide /usr/bin/orbd (orbd) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/servertool to provide /usr/bin/servertool (servertool) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/tnameserv to provide /usr/bin/tnameserv (tnameserv) in auto mode
Setting up openjdk-8-jdk-headless:amd64 (8u402-ga-2ubuntu1~22.04) ...
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/clhsdb to provide /usr/bin/clhsdb (clhsdb) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/extcheck to provide /usr/bin/extcheck (extcheck) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/hsdb to provide /usr/bin/hsdb (hsdb) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/idlj to provide /usr/bin/idlj (idlj) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/javah to provide /usr/bin/javah (javah) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jhat to provide /usr/bin/jhat (jhat) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jsadebugd to provide /usr/bin/jsadebugd (jsadebugd) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/native2ascii to provide /usr/bin/native2ascii (native2ascii) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/schemagen to provide /usr/bin/schemagen (schemagen) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/wsgen to provide /usr/bin/wsgen (wsgen) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/wsimport to provide /usr/bin/wsimport (wsimport) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/xjc to provide /usr/bin/xjc (xjc) in auto mode
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link
```

In [2]:
```
!sudo apt update
```

```
Get:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease [3,626 B]
Get:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  InRelease [1,581 B]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:5 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  Packages [814 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:7 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy InRelease
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1,748 kB]
Hit:9 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Hit:10 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy InRelease
Hit:11 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:12 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Get:13 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [2,251 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1,077 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2,032 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,369 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2,333 kB]
Fetched 11.9 MB in 3s (4,386 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
45 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

In [3]:
```
#  Import the libraries we will need
import pyspark
from pyspark.sql import *
from pyspark.sql.functions import *
from pyspark import SparkContext, SparkConf

# Create Spark session and ContextRun PySpark.
# create the session
conf = SparkConf().set("spark.ui.port","4050")
# create the context
sc = pyspark.SparkContext(conf=conf)
spark = SparkSession.builder.appName("DataFrame").config('spark.ui.port', '4050').getOrCreate()
spark
```

Out[3]:

**SparkSession - in-memory**

**SparkContext**

[Spark UI (http://97f0daf8320a:4050)](http://97f0daf8320a:4050)

**Version**

`v3.5.1`

**Master**

`local[*]`

**AppName**

`pyspark-shell`

In [83]:
```
# Q2: Write efficient Spark code for creating K-shingles given a huge document and K as input.
import hashlib
HugeDocument = sc.parallelize(['D1,Pretend it is docu','D2,Pretend it no docu'])
K=9
d=0

def hashIt(shingle):
    # Hash the shingle to 4 bytes
    buckets = 8
    hashObj = hashlib.sha256(shingle.encode())
    hashed = int.from_bytes(hashObj.digest(), byteorder='big') % buckets
    return hashed
def getShingles(line):
    global K
    documentNumber, text = line.split(',')
    text = text.lower()
    setOfShingles = set()
    for i in range(len(text) - K + 1):
        shingle = text[i:i+K]
        hashedShingle = hashIt(shingle)
        setOfShingles.add(hashedShingle)
    return documentNumber, setOfShingles

shingles=HugeDocument.map(lambda x: getShingles(x))

# Removing Duplicate Shingles
uniqueShingles = shingles.flatMap(lambda x: x[1]).distinct().collect()

# Getting Boolean Matrix ---->

shingleIndex = {shingle: i for i, shingle in enumerate(uniqueShingles)} # Dictionary to map each shingle to an index
docShin = shingles.map(lambda x: (x[0], [shingleIndex[shingle] for shingle in x[1]])) # List of document and shingle pairs
sparseM = docShin.flatMapValues(lambda x: x).map(lambda x: (x, 1)).reduceByKey(lambda x, y: x).sortByKey()
dfData = sparseM.map(lambda x: (x[0][0], x[0][1], x[1])).toDF(["Document", "Shingle", "Value"]) # Convert sparseM RDD to DataFrame
pivotedDf = dfData.groupby("Document").pivot("Shingle").agg({"Value": "max"})
pivotedDf = pivotedDf.fillna(0) # Null values filled with 0
pivotedDf.show()
```

```
+--------+---+---+---+---+---+---+---+---+
|Document|  0|  1|  2|  3|  4|  5|  6|  7|
+--------+---+---+---+---+---+---+---+---+
|      D1|  1|  1|  1|  0|  1|  1|  1|  1|
|      D2|  1|  0|  1|  1|  1|  1|  1|  0|
+--------+---+---+---+---+---+---+---+---+
```

In [84]:
```
# Q3:  Write an efficient SPARK code for Minhashing
import random

# 100 random permutations of the rows
KPermutations = [random.sample(range(len(uniqueShingles)), len(uniqueShingles)) for _ in range(100)]

def updateSignature(row, permutations):
    document, shingleIndices = row
    updatedSig = [float('inf')] * len(permutations)
    for shingleIndex in shingleIndices:
        for i, perm in enumerate(permutations):
            if perm.index(shingleIndex) < updatedSig[i]:
                updatedSig[i] = perm.index(shingleIndex)
    return document, updatedSig

# Update signature matrix with min-hash
signatureMatrix = docShin.map(lambda x: updateSignature(x, KPermutations))
signatureDF = signatureMatrix.toDF(["Document", "Signature"])
signatureDF.show(truncate=False)
```

```
+--------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------+
|Document|Signature
|
+--------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------+
|D1      |[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0]|
|D2      |[1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 2, 0, 0, 1,
0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 2, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]|
+--------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------+
```

In [ ]: