```
#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'
```

```
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=ef34d81303fcc18326e1bcf553d6dc1d0cd1b859cd8d45dc35c021871269f1ba
  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 ... 121918 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_2_5.so.3 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

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

```
#  Importing Required Libraries
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
```

SparkSession - in-memory

SparkContext

Spark UI

Version
`v3.5.1`
Master
`local[*]`
AppName
`pyspark-shell`

- Write a Spark code that read a directed graph as adjacency list from a textFile and convert it to undirected graph.

```
inp = sc.textFile('input.txt').map(lambda x: (x.split('->')[0],x.split('->')[1].split(',')))
inp1=inp.flatMap(lambda x: [(x[0],y) for y in x[1]])
inp2=inp.flatMap(lambda x: [(y,x[0]) for y in x[1]])
inpAll=inp1.union(inp2)
results=inpAll.groupByKey()
for key, value in results.collect():
    print(key, list(value))
```

```
4 ['3', '2']
3 ['1', '4']
1 ['2', '3']
2 ['4', '1']
```

Try Before Final Exam

```
directedGraph=sc.textFile('input.txt')
vertexEdges=directedGraph.map(lambda line: (line.split('->')[0],line.split('->')[1].split(',')) )
vertexEdgesAll=vertexEdges.flatMap(lambda value:[(val,value[0]) for val in value[1]])
vertexEdgesAll2=vertexEdges.flatMap(lambda value:[(value[0],val) for val in value[1]])
res=vertexEdgesAll.union(vertexEdgesAll2).groupByKey().collect()
for key, value in res:
    print(key, list(value))
```

```
4 <pyspark.resultiterable.ResultIterable object at 0x7cc85396bd00>
3 <pyspark.resultiterable.ResultIterable object at 0x7cc85396b6d0>
1 <pyspark.resultiterable.ResultIterable object at 0x7cc853898d60>
2 <pyspark.resultiterable.ResultIterable object at 0x7cc85389b490>
```

```
# Read the directed graph from the input file
directedGraph = sc.textFile('input.txt')

# Parse each line into a tuple of (vertex, [edges])
vertexEdges = directedGraph.map(lambda line: (line.split('->')[0].strip(), line.split('->')[1].strip().split(',')))

# Create bidirectional (vertex, edge) pairs
bidirectionalEdges = vertexEdges.flatMap(lambda value: [(value[0], val.strip()) for val in value[1]] + [(val.strip(), val
# bidirectionalEdges.collect()
# Group by vertex and aggregate the edges
undirectedGraph = bidirectionalEdges.groupByKey().mapValues(lambda vals: sorted(set(vals))).collect()

# Print the results
for key, value in undirectedGraph:
    print(f"{key}-> {', '.join(value)}")
```

```
1-> 2, 3
4-> 2, 3
2-> 1, 4
3-> 1, 4
```