```
#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 2.6 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=3929933b3a17babbbbc2750a44f3ecbc218cc0347965def37072fb39560be322
  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 ... 121920 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/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/libtbbmalloc.so.2 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

Adjacency List to Edge List Conversion

```
# Read the file and create the RDD
rdd1 = sc.textFile('file.txt').map(lambda line: (line.split('->')[0], line.split('->')[1].split(',')))
Rdd = rdd1.flatMap(lambda x: [(x[0],y) for y in x[1]])

# Collect the results
Rdd.collect()
```

```
[('1', '2'),
 ('1', '4'),
 ('2', '3'),
 ('2', '4'),
 ('3', '1'),
 ('4', '0'),
 ('5', '6'),
 ('5', '2'),
 ('6', '3'),
 ('7', '2')]
```

Edge List to Adjacency List Conversion

```
# Read the file and create the RDD
rdd1 = sc.textFile('file2.txt').map(lambda line: (line.split(',')[0], line.split(',')[1]))
rdd2=rdd1.groupByKey().collect()

for vertex,neigbours in rdd2:
  print('\n',vertex,'-> ',end='')
  for neigbour in neigbours:
    print(neigbour,end=' ')
```

```
 1 -> 2 4
 4 -> 0
 2 -> 3 4
 3 -> 1
 5 -> 6 2
 6 -> 3
 7 -> 2
```

Page Rank Simple Algorithm (Not sure if correct)

```
rdd=sc.textFile('file3.txt').map(lambda line: (line.split('->')[0],line.split('->')[1].split(',')))
#  First Type of Packet Generation (Neighbours of Node)
rddP1=rdd.map(lambda point: (point[0].split(',')[0],point[1]))
print('First Type of Packet Generation (Neighbours of Node)')
print(rddP1.collect())
#  Second Type of Packet Generation (Importance of Node)
rddP2=rdd.flatMap(lambda point: [(p,float(point[0].split(',')[1])/len(point[1])) for p in point[1]])
print('Second Type of Packet Generation (Importance of Node)')
print(rddP2.collect())
rddP2Sum=rddP2.reduceByKey(lambda x,y : x+y)
print('Total Importance of Each Node')
rddP2Sum.collect()
```

```
First Type of Packet Generation (Neighbours of Node)
[('1', ['2', '4']), ('2', ['3', '4']), ('3', ['1']), ('4', ['0']), ('5', ['6', '2']), ('6', ['3']), ('7', ['2'])]
Second Type of Packet Generation (Importance of Node)
[('2', 0.4), ('4', 0.4), ('3', 0.4), ('4', 0.4), ('1', 0.8), ('0', 0.8), ('6', 0.4), ('2', 0.4), ('3', 0.8), ('2', 0.8)]
Total Importance of Each Node
[('4', 0.8),
 ('1', 0.8),
 ('0', 0.8),
 ('2', 1.6),
 ('3', 1.2000000000000002),
 ('6', 0.4)]
```

```
# Chatgpt code
# Read data from file and create the RDD
rdd = sc.textFile('file3.txt').map(lambda line: (line.split('->')[0], line.split('->')[1].split(',')))

# Initial PageRank values for each node
initial_rank = 1.0

# Damping factor for PageRank calculation
damping_factor = 0.85

# Number of iterations for PageRank computation
iterations = 10

# Function to compute contributions of neighbors
def compute_contributions(neighbors, rank):
    num_neighbors = len(neighbors)
    for neighbor in neighbors:
        yield (neighbor, rank / num_neighbors)
```

```
# Iterative PageRank computation
for i in range(iterations):
    # First type of packet generation: Neighbors of each node
    rdd_neighbors = rdd.map(lambda point: (point[0].split(',')[0], point[1]))
    print('First Type of Packet Generation (Neighbors of Node)')
    print(rdd_neighbors.collect())

    # Second type of packet generation: Importance of each node
    rdd_importance = rdd.flatMap(lambda point: compute_contributions(point[1], float(point[0].split(',')[1])))
    print('Second Type of Packet Generation (Importance of Node)')
    print(rdd_importance.collect())

    # Summing up the importance of each node
    rdd_sum = rdd_importance.reduceByKey(lambda x, y: x + y)
    print('Total Importance of Each Node after Iteration', i+1)
    print(rdd_sum.collect())

    # Update PageRank values for the next iteration
    rdd = rdd_neighbors.join(rdd_sum).mapValues(lambda x: (x[0], damping_factor * x[1] + (1 - damping_factor)))

# Collect final PageRank values
final_ranks = rdd.collect()
print('Final PageRank Values:')
print(final_ranks)
```

```
First Type of Packet Generation (Neighbors of Node)
[('1', ['2', '4']), ('2', ['3', '4']), ('3', ['1']), ('4', ['0']), ('5', ['6', '2']), ('6', ['3']), ('7', ['2'])]
Second Type of Packet Generation (Importance of Node)
[('2', 0.4), ('4', 0.4), ('3', 0.4), ('4', 0.4), ('1', 0.8), ('0', 0.8), ('6', 0.4), ('2', 0.4), ('3', 0.8), ('2', 0.8)]
Total Importance of Each Node after Iteration 1
[('4', 0.8), ('1', 0.8), ('0', 0.8), ('2', 1.6), ('3', 1.2000000000000002), ('6', 0.4)]
First Type of Packet Generation (Neighbors of Node)
[('4', (['0'], 0.8300000000000001)), ('3', (['1'], 1.17)), ('6', (['3'], 0.49000000000000005)), ('1', (['2', '4'], 0.8300000000000001)), ('2', (['3', '4'], 1.5100000000000002
Second Type of Packet Generation (Importance of Node)
---------------------------------------------------------------------------
Py4JJavaError                             Traceback (most recent call last)
<ipython-input-45-e37d8b4f0b2e> in <cell line: 21>()
     28     rdd_importance = rdd.flatMap(lambda point: compute_contributions(point[1], float(point[0].split(',')[1])))
     29     print('Second Type of Packet Generation (Importance of Node)')
---> 30     print(rdd_importance.collect())
     31
     32     # Summing up the importance of each node

                                  3 frames
/usr/local/lib/python3.10/dist-packages/py4j/protocol.py in get_return_value(answer, gateway_client, target_id, name)
    324             value = OUTPUT_CONVERTER[type](answer[2:], gateway_client)
    325             if answer[1] == REFERENCE_TYPE:
--> 326                 raise Py4JJavaError(
    327                     "An error occurred while calling {0}{1}{2}.\n".
    328                     format(target_id, ".", name), value)

Py4JJavaError: An error occurred while calling z:org.apache.spark.api.python.PythonRDD.collectAndServe.
: org.apache.spark.SparkException: Job aborted due to stage failure: Task 1 in stage 69.0 failed 1 times, most recent failure: Lost task 1.0 in stage 69.0 (TID 137)
(a6219f99ed85 executor driver): org.apache.spark.api.python.PythonException: Traceback (most recent call last):
  File "/usr/local/lib/python3.10/dist-packages/pyspark/python/lib/pyspark.zip/pyspark/worker.py", line 1247, in main
    process()
  File "/usr/local/lib/python3.10/dist-packages/pyspark/python/lib/pyspark.zip/pyspark/worker.py", line 1239, in process
    serializer.dump_stream(out_iter, outfile)
  File "/usr/local/lib/python3.10/dist-packages/pyspark/python/lib/pyspark.zip/pyspark/serializers.py", line 274, in dump_stream
    vs = list(itertools.islice(iterator, batch))
  File "/usr/local/lib/python3.10/dist-packages/pyspark/python/lib/pyspark.zip/pyspark/util.py", line 83, in wrapper
    return f(*args, **kwargs)
  File "<ipython-input-45-e37d8b4f0b2e>", line 28, in <lambda>
IndexError: list index out of range

        at org.apache.spark.api.python.BasePythonRunner$ReaderIterator.handlePythonException(PythonRunner.scala:572)
        at org.apache.spark.api.python.PythonRunner$$anon$3.read(PythonRunner.scala:784)
        at org.apache.spark.api.python.PythonRunner$$anon$3.read(PythonRunner.scala:766)
        at org.apache.spark.api.python.BasePythonRunner$ReaderIterator.hasNext(PythonRunner.scala:525)
        at org.apache.spark.InterruptibleIterator.hasNext(InterruptibleIterator.scala:37)
        at scala.collection.Iterator.foreach(Iterator.scala:943)
        at scala.collection.Iterator.foreach$(Iterator.scala:943)
        at org.apache.spark.InterruptibleIterator.foreach(InterruptibleIterator.scala:28)
        at scala.collection.generic.Growable.$plus$plus$eq(Growable.scala:62)
        at scala.collection.generic.Growable.$plus$plus$eq$(Growable.scala:53)
        at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:105)
        at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:49)
        at scala.collection.TraversableOnce.to(TraversableOnce.scala:366)
        at scala.collection.TraversableOnce.to$(TraversableOnce.scala:364)
        at org.apache.spark.InterruptibleIterator.to(InterruptibleIterator.scala:28)
        at scala.collection.TraversableOnce.toBuffer(TraversableOnce.scala:358)
        at scala.collection.TraversableOnce.toBuffer$(TraversableOnce.scala:358)
        at org.apache.spark.InterruptibleIterator.toBuffer(InterruptibleIterator.scala:28)
        at scala.collection.TraversableOnce.toArray(TraversableOnce.scala:345)
        at scala.collection.TraversableOnce.toArray$(TraversableOnce.scala:339)
        at org.apache.spark.InterruptibleIterator.toArray(InterruptibleIterator.scala:28)
        at org.apache.spark.rdd.RDD.$anonfun$collect$2(RDD.scala:1049)
        at org.apache.spark.SparkContext.$anonfun$runJob$5(SparkContext.scala:2438)
        at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:93)
        at org.apache.spark.TaskContext.runTaskWithListeners(TaskContext.scala:166)
        at org.apache.spark.scheduler.Task.run(Task.scala:141)
        at org.apache.spark.executor.Executor$TaskRunner.$anonfun$run$4(Executor.scala:620)
        at org.apache.spark.util.SparkErrorUtils.tryWithSafeFinally(SparkErrorUtils.scala:64)
```

```
        at org.apache.spark.util.SparkErrorUtils.tryWithSafeFinally$(SparkErrorUtils.scala:61)
        at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:94)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:623)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:750)

Driver stacktrace:
        at org.apache.spark.scheduler.DAGScheduler.failJobAndIndependentStages(DAGScheduler.scala:2856)
        at org.apache.spark.scheduler.DAGScheduler.$anonfun$abortStage$2(DAGScheduler.scala:2792)
        at org.apache.spark.scheduler.DAGScheduler.$anonfun$abortStage$2$adapted(DAGScheduler.scala:2791)
        at scala.collection.mutable.ResizableArray.foreach(ResizableArray.scala:62)
        at scala.collection.mutable.ResizableArray.foreach$(ResizableArray.scala:55)
        at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:49)
        at org.apache.spark.scheduler.DAGScheduler.abortStage(DAGScheduler.scala:2791)
        at org.apache.spark.scheduler.DAGScheduler.$anonfun$handleTaskSetFailed$1(DAGScheduler.scala:1247)
```