5 GraphX

For this section, you will be assessing the same harbour data as in the Hadoop Graph Processing section, but we have also provided a mirrored version to allow you to treat this as an undirected graph. You may use either version of the data as appropriate. The following documentation will be very useful for you: **GraphOps Documentation** , **GraphX coding guide**. For all of the following, your answer need not be a single command - you may create outputs and perform follow-up operations on those outputs, but GraphX should be the "driver" in your answers - e.g. an answer that exclusively uses SparkSQL unions (and no GraphX operations) to make links would be undesirable. Download the data using the following code (remember to make it all one line!). You may use whichever data is preferable for any given question!

For the mirrored dataset:

```
wget --no-check-certificate
 ↪ 'https://docs.google.com/uc?export=download&id=19Uubzr_jcXGiVse5EJCOmBpLNEHH7YXY'
 ↪ -O hadoop_mirrored.csv
```

For a dataset of edges:

```
wget --no-check-certificate
 ↪ 'https://docs.google.com/uc?export=download&id=1QmpOehpXOWGyUZ6AlNpmBrVxY9tpOLmj'
 ↪ -O hadoop_mirrored.csv
```

1. Import the data and create a graph representing the data

2. Generate an array of each harbour's connected routes - consult the spark documentation to identify the most suitable method for this. Alternatively, you may use Spark commands to generate this information.

3. Which harbour(s) is/are served by route "Porium_Thirty-one"?

4. Which harbour has the most routes associated with it (NB - please return a count of ROUTES PER HARBOUR for this) . If there are multiple harbours with the same number of routes, list them all.

5. Which harbour is connected to the most other harbours - for this, you can transform your earlier outputs, or use a new GraphX method. If there are multiple harbours with the same number of connections, list them all. You may find the above edge data useful! (NB - please return a count of NEIGHBOURING HARBOURS PER HARBOUR for this - if your graph is structured as harbours as nodes and routes as edges, please return a count of neighbouring nodes etc. - do not just return a count of edges, even if they are the same number. Please consult the GraphOps documentation above for tools that may be useful to you!).