Q. Explain the different join strategies in Spark and which strategy you will be adopting when joining Parquet File 1 and 2 if you are implementing the code in Spark Dataframe. Your answer can be saved as joins.pdf under the main repository.

Spark reads data as a partition. To ensure join functions, spark can shuffle the data such that similar data are group within the same partition.

In shuffle hash, after the data is shuffled, smaller data are hashed. The smaller data will then be matched with the main data and joined.

In sort merge, both data sets are shuffled and sorted. Sort enables the keys to be sorted In order such that during the join phase, it only needs to search through the dataset until it is no longer the same. This will be useful if both data sets are large.

In broadcast join, the smaller dataset is broadcasted to all the executors. The data in the big dataset is then hashed and joined. The default size value for smaller dataset is 10mb to 8gb. Any larger might result in out of memory error.

With File 2 being a reference table, a broadcast hash join would make the most sense since dataset B is small.

However, in my implementation, all operations are performed on RDDs. The join strategies above can only be implemented on dataframes, which meant that I would need to convert them into .dataframe. Furthermore, considering the size of the dataset is small(1mil and 10,000 rows), a basic inner join would be sufficient for this operation.