Question :

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.

1. Broadcast hash join
    a) By default, spark will consider this join function if one of the datasets is less than 10 MB (can be configured using spark.sql.autoBroadcastJoinThreshold)
    b) The smaller dataset will be broadcast to all the executor nodes and stored on driver node
2. Shuffle hash join
    a) Is use when both tables are more than 10MB
    b) The function will be repartitioning on both tables based on the joining keys using hash partitioning and the join will based on hash tabled been created for lookup activity.
3. Sort merge join
    a) Better performance compares to shuffle hash join when working with both large dataset
    b) The function will repartition both tables based on joining keys using hash partitioning, sorting data happen on each partition individually follow by merging base on the joining key
4. Cartesian join
    a) every record from one table being joined with every row in other table
    b) very costly in term of memory and network usage and might lead to out of memory issue when there is insufficient space to execute the join
5. Broadcast nested loop join
    a) Is use when one of the datasets is small and another dataset is large
    b) Each dimension in the smaller table will be relates to all the records from large table
    c) this dimension data will be broadcasted to all executors handling portions of the large table data

Broadcast hash join is more suitable to join parquet file 1 and 2 because dataset b is considered small dataset and dataset a is consider big dataset. So spark can broadcast dataset b to all worker nodes allowing each partition of dataset a to perform a local join with broadcasted data.