Data Engineering

- Total points = 100.
- State all assumptions. For questions where descriptive solutions are required, you will be graded both on the correctness and clarity of your reasoning.

Question 1 **Points 10**

Explain the steps to produce a single sorted list given two sorted lists $R(A)$ and $S(A)$. Do not eliminate duplicates.

Question 2 **Points 10**

Consider a large table $R(A, B)$ that is stored as blocks across many machines on a distributed file-system like the Hadoop Distributed File System (HDFS). Explain the steps to sort the tuples in $R(A, B)$ in parallel using all the machines. The sorted result should be written back to the distributed file system. Feel free to choose how you will store the final output, but make it clear in your answer why you made this choice.

Question 3 **Points 10**

In class we talked about Sort Merge join. Explain the steps to produce the join result of two tables $R(A, B)$ and $S(A, C)$ using Sort Merge join on $R.A = S.A$. Illustrate your answer using the following tuples in R and S:

```
R = {(1,a),(3,c),(2,b),(1,d),(2,a),(3,d),(1,d),(4,a),(1,c),(3,a)}
S = {(1,f),(4,c),(2,b),(1,g),(3,a)}
```

Question 4 **Points 20**

1. Explain the steps involved in a map-side join of two tables R and S.
2. Explain the steps involved in a reduce-side join of two tables R and S.
3. Describe the conditions when a map-side join of R and S will be preferable to a reduce-side join of R and S.
4. Describe the conditions when a reduce-side join of R and S will be preferable to a map-side join of R and S.

Question 5 **Points 10**

Give an example of how sorting can be used to great advantage while doing a reduce-side join. State clearly where the benefits arise (e.g., join runs faster, join uses less memory, etc.) and why.

Question 6 **Points 10**

Suppose one key has many duplicates in two tables that need to be joined. For example, consider the following tables $R(A, B)$ and $S(A, C)$ which have to be joined on $R.A = S.A$.

```
R = {(1,a),(3,c),(2,b),(1,d),(2,a),(1,d),(1,d),(4,a),(1,c),(3,a)}
S = {(1,f),(1,c),(2,b),(1,g),(1,a)}
```

Notice how the key "1" has many duplicates in attribute R.A as well as in S.A.

1. How will the presence of such duplicates affect the performance of a Sort Merge join?
2. How will the presence of such duplicates affect the performance of a Tuple Nested Loop join?
3. How will the presence of such duplicates affect the performance of a map-side join?
4. How will the presence of such duplicates affect the performance of a reduce-side join?

Question 7 **Points 30**

Consider the following SQL query:

```
Select *
From  users, clicks, geoinfo
Where users.name = clicks.user and geoinfo.ipaddr = users.ipaddr
```

The following information is available:

- geoinfo has 1000 tuples. Each tuple has a size of 100 bytes.
- users has 1 Million (10^6) tuples. Each tuple has a size of 100 bytes.
- clicks has 100 Million (10^8) tuples. Each tuple has a size of 100 bytes.
- clicks is stored sorted on the clicks.user attribute.
- Every tuple in clicks will join with exactly one tuple in users. Every tuple in users will join with exactly one tuple in geoinfo.
- There is a clustered index on the clicks.user attribute that enables quick lookup and fetching of all the tuples in clicks for any value of clicks.user.

1. Describe how the Selinger algorithm will work in order to find the best physical plan for the above SQL query. Assume that three join operators (sort merge join, tuple nested loop join that uses an index on the right side table, and tuple nested loop join that uses a full table scan on the right side table) and two scan operators (full table scan and index scan) are available. Your answer must include details like: (i) interesting orders, (ii) the lattice and the physical plans considered at each node in the lattice, (iii) your estimate of the best plan picked for each node in the lattice.
2. In class, we studied how the Selinger algorithm considers only left-deep join trees. Describe how you will extend the Selinger algorithm to consider only right-deep join trees. (See Figure 1 for examples of join trees.)
3. Describe how you will extend the Selinger algorithm to consider all bushy join trees. (See Figure 1 for examples of join trees.)

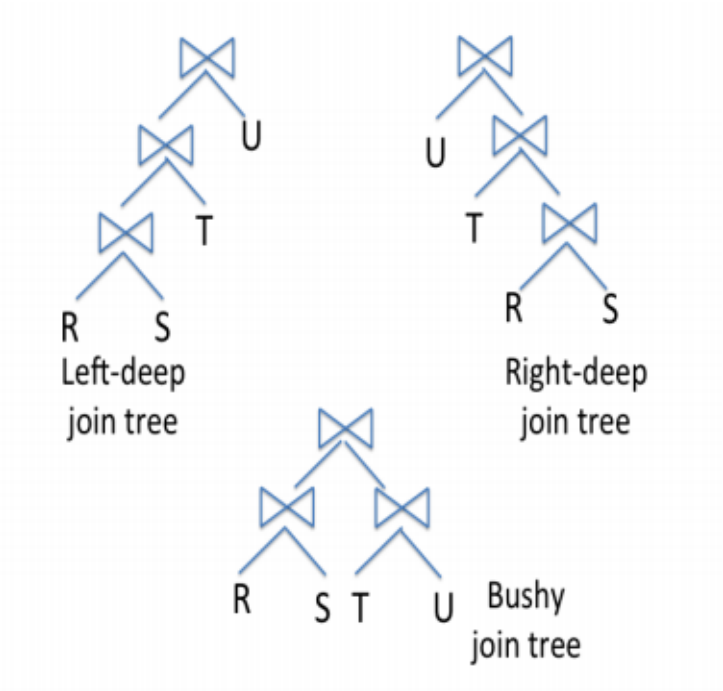

Figure 1: Example join trees