CS150A Quiz #5

Query Optimization

Assume that the optimizer follows a System-R style implementation for all applicable questions.

1. **Q 1: T/F - If a term has a large reduction factor, the output of the query will have more tuples than if it had a small reduction factor.** *
   Mark only one oval.
   - ◯ True
   - ◯ False

2. **Q 2: T/F - An equidepth histogram gives better resolution on low-frequency entries than a equiwidth histogram.** *
   i.e. it gives more detailed information for these entries.
   Mark only one oval.
   - ◯ True
   - ◯ False

3. **Q 3: When doing a cross join on tables A, B, C, and D, which of the following query plans do we consider?** *
   Mark all that apply.
   Check all that apply.
   - ☐ None of the above
   - ☐ (A join (B join C)) join D
   - ☐ A join ((B join C) join D)
   - ☐ ((A join B) join C) join D
   - ☐ A join (B join (C join D))
   - ☐ (A join B) join (C join D)

4. **Q4: Which of the following access or join methods will result in an interesting order in a query where we require the output to be sorted?** *

Check all that apply.

- [ ] File scan
- [ ] Sort-Merge Join
- [ ] Block-Nested Loops Join
- [ ] Clustered Index Traversal
- [ ] Hash Join

Suppose that we have three tables, R, S, and T. We are running the following query:

```
SELECT *
FROM R, S, T
WHERE R.a = S.a
AND S.b = T.b;
```

Assume that our database has no indices and that none of the relations are sorted in any interesting or useful way. Since we only have one possible single-table access method for each table, we ignore the costs of accessing a single table.

Assume that all provided join costs are for the optimal join algorithm for that join.

These are the two-table join costs:
1) S join R = 2,000
2) R join S = 6,000
3) R join T = 5,000
4) T join R = 1,000
5) T join S = 3,000
6) S join T = 4,000

5. **Q 5: Which of the above two-table join plans will be selected?** *

Check all that apply.

- [ ] 1
- [ ] 2
- [ ] 3
- [ ] 4
- [ ] 5
- [ ] 6

We now add the third table and have the following join costs:
1) (R join S) join T = 10,000
2) T join (R join S) = 6,000
3) R join (S join T) = 12,000
4) T join (S join R) = 11,000
5) (R join T) join S = 10,000
6) S join (R join T) = 7,000

7) (T join R) join S = 14,000
8) S join (T join R) = 16,000
9) (S join T) join R = 13,000
10) (S join R) join T = 15,000
11)(T join S) join R = 20,000
12) R join (T join S) = 9,000

6. **Q 6: Which of these will the optimizer select as your final query plan?** *

Mark only one oval.

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12