<u>ASSIGNMENT 1</u>

COMP3323: Advanced database systems

Answer 1.
Relation r has 50000 tuples
25 tuples of r fit in one block

Relation s has 6000 tuples
60 tuples of s fit in one block

System has a memory of M = 100 blocks.

i)

a) *Nested-loop join*:

Cost = $\frac{50000}{25} + 50000(\frac{6000}{60})$

= 5,002,000

b) *Block nested-loop join*:

Cost = $\frac{50000}{25} + (\frac{50000}{25} \div (100-2))(\frac{6000}{60})$

= 4,100

c) *Merge-Join (external sorting has been already performed):*

Cost = $\frac{50000}{25} + (\frac{6000}{60})$

= 2,100

d) *Hash-Join:*

Cost = $3 * (\frac{50000}{25} + \left(\frac{6000}{60}\right))$

= 6,300

ii) Number of bytes for header is 16. Sibling pointer, record-id pointer, key value pointers are 12. Block size is 4096 bytes.

= $\lfloor\frac{4096-96-12}{12+12}\rfloor$

= 166

Total is 167 as 166 + 1

First Case is S as outer relation

Height of B+ tree on r.B = $1 + log_{167}(50000/166) = 3$

= 100 + 3*(6000) + 30000

= 48,100

Second Case is R as outer relation

Height of B+ tree on s.B = $1 + log_{167}(6000/166) = 2$

= 2000 + 2*(50000) + 30000

= 132,000

As the cost is bigger for Second case than First case, **minimum cost is 48,100**.

Answer 2.

a)

Going bottom up in the query plan.

The cost of file scan (assuming linear scan) for $\sigma_{71 \leq R.a \leq 80} = \frac{1000}{10} = 100$

The cost of file scan (assuming linear scan) for $\sigma_{S.b < 5} = \frac{10000}{10} = 1000$

Knowing that the records are uniformly distributed,

The cost of writing to $T_1 = \frac{80-71+1}{200-1+1} * \frac{1000}{10} = 5$

The cost of writing to $T_2 = \frac{1}{10-1+1} * \frac{10000}{10} = 100$

The cost of block nested loop join = 505

Hence, the total cost of block accessed is 1710.

b) c is common attribute for R or S.

If we assume every tuple/record in R produces R ⋈ S, then the output size is:

$= \frac{50*1000}{10} = 5{,}000$

As V(c, R) and V(c, S) are the same (denominator in the previous equation), the **output size is 5,000.**

Answer 3

ii)

Trying out random inputs, number of tuples we get:

<u>Input 5 25</u>

- Estimated result equi-width histogram: 2831.2

- Estimated result equi-depth histogram: 2825.1315789473683
- Real result: 2623

<u>Input 5 5</u>

- Estimated result equi-width histogram: 45.0
- Estimated result equi-depth histogram: 66.47368421052632
- Real result: 45

<u>Input 24 58</u>

- Estimated result equi-width histogram: 6970.2
- Estimated result using the equi-depth histogram: 6630.75
- Real result: 7118

<u>Input 75 79</u>

- Estimated result equi-width histogram: 72.5
- Estimated result equi-depth histogram: 225.53571428571428
- Real result: 49

<u>Input 39 45</u>

- Estimated result equi-width histogram: 1412.4
- Estimated result equi-depth histogram: 1443.4285714285713
- Real result: 1440

<u>Input 79 79</u>

- Estimated result equi-width histogram: 14.5
- Estimated result equi-depth histogram: 45.107142857142854
- Real result: 12

From comparing both histogram's result, it can be concluded that **equi-width is closer to the real result** than equi-depth most of the time. This is because data distribution is highly skewed eg. 24 to 58 (nearly half the range of values) have 7,118 tuples (nearly 70% of the tuples).