CS3223: Database Management Systems
Tutorial 7
(Week of 14th March, 2022)

1. Below are the vital statistics for three relations, W, X and Y:

W(a,b)	**X(b, c)**	**Y(c, d)**
T(W) = 100	T(X) = 200	T(Y) = 300
V(W, a) = 20	V(X, b) = 50	V(Y, c) = 50
V(W, b) = 60	V(X, c) = 100	V(Y, d) = 50

   Estimate the sizes of relations that are results of the following expressions:
   - $\sigma_{a=10}(W)$
   - $\sigma_{a>10}(W)$
   - $\sigma_{a=10 \wedge b=2}(W)$
   - $\sigma_{a=10 \wedge b>2}(W)$
   - $\sigma_{c=20}(X) \bowtie Y$
   - $W \bowtie X \bowtie Y$

2. Consider a database with three relations: R(a, b, c), S(c, b, d) and T(d, b, e). The primary keys of relations R, S and T are attributes a, c and d respectively. Suppose R has 500 tuples, S has 500 tuples and T has 1,000 tuples. Let the number of distinct values of b be 100, i.e., V(R,b) = V(S, b) = V(T, b) = 100. Suppose every attribute is of the same size, and each page can hold 10 tuples of R. Suppose the database maintains histograms for attribute b by keeping ONLY the frequency of the THREE most common values, as tabulated below (the other values are not maintained and are assumed to be uniformly distributed). Estimate the size of this query in terms of pages and tuples:

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

	0	**1**	**2**	**3**	**4**	**Others**
R.b	72	30	10			388
S.b		40	16	250		194
T.b	100		7		20	873

3. Consider the query: $R1(a,b) \bowtie R2(b, c) \bowtie R3(c, d) \bowtie R4(a, d)$. Let T(R) denote number of tuples of table R, and V(R,a) denote the number of distinct value of attribute a. Suppose T(R1) = T(R4) = 1000 and T(R2) = T(R3) = 100. Further, assume that V(R1,a) = V(R1,b) = V(R2,b) = V(R3,d) = V(R4,d) = V(R4,a) = 100, and V(R2,c) = V(R3,c) = 10. Consider a greedy algorithm that restricts its space to left-deep trees only, and greedily expands the join plan by keeping the intermediate results as small as possible at each level of the tree. For simplicity, let's ignore the join algorithms, and use the number of intermediate result tuples as the cost metric, i.e., we want to minimize the total intermediate result tuples produced.
   a. What is the order selected by the greedy algorithm? What is its cost?
   b. What is the globally optimal join ordering and its cost?

4. (Question by Magdalena Balazinska) Consider the following SQL query that finds all applicants who want to major in CSE, live in Seattle, and go to a school ranked better than 10 (i.e., rank < 10).

Relation	Cardinality	Number of pages	Primary key
Applicants(id, name, city, sid)	2,000	100	id
Schools(sid, sname, srank)	100	10	sid
Major(id, major)	3,000	200	(id,major)

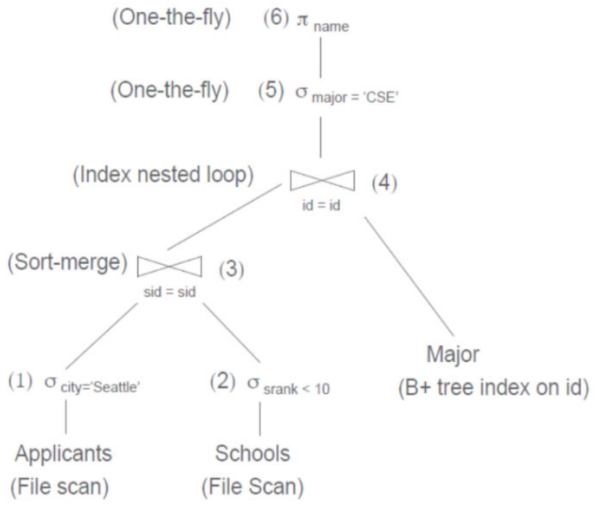
‘

SELECT A.name
FROM Applicants A, Schools S, Major M
WHERE A.sid = S.sid AND A.id = M.id AND A.city = ’Seattle’
AND S.rank < 10 AND M.major = ’CSE’

And assuming:

- Each school has a unique rank number (srank value) between 1 and 100.
- There are 20 different cities.
- Applicants.sid is a foreign key that references Schools.sid.
- Major.id is a foreign key that references Applicants.id.
- There is an unclustered, secondary B+ tree index on Major.id and all index pages are in memory.
- You can assume sufficient buffer size or use buffer size of 10 pages.

(a) What is the cost of the query plan below? Count only the number of page I/Os.



(b) The System R optimizer uses a dynamic programming algorithm coupled with a set of heuristics to enumerate query plans and limit its search space. Draw two query plans for the above query that the optimizer would NOT consider. For each query plan, indicate why it would not be considered.