1. Consider a relation $R(a, b, c, d, e)$ containing 5,000,000 records, where each data page of the relation holds 10 records. $R$ is organized as a sorted file with secondary indexes. Assume that $R.a$ is a candidate key for $R$, with values lying in the range 0 to 4,999,999, and that $R$ is stored in $R.a$ order. For each of the following relational algebra queries from (a) to (f), state which of the following approaches (or combination thereof) is most likely to be the cheapest:

- Access the sorted file for $R$ directly.
- Use a clustered B+ tree index on attribute $R.a$.
- Use a linear hashed index on attribute $R.a$.
- Use a clustered B+ tree index on attributes ($R.a, R.b$).
- Use a linear hashed index on attributes ($R.a, R.b$).
- Use an unclustered B+ tree index on attribute $R.b$.

(a) $\sigma_{a<50,000 \land b<50,000}(R)$
(b) $\sigma_{a=50,000 \land b<50,000}(R)$
(c) $\sigma_{a>50,000 \land b=50,000}(R)$
(d) $\sigma_{a=50,000 \land b=50,010}(R)$
(e) $\sigma_{a\neq50,000 \land b=50,000}(R)$
(f) $\sigma_{a<50,000 \land b=50,000}(R)$

2. Consider the join $R \bowtie_{R.a=S.b} S$, given the following information about the relations to be joined. The cost metric is the number of page I/Os unless otherwise noted, and the cost of writing out the result should be uniformly ignored.

Relation $R$ contains 10,000 tuples and has 10 tuples per page.
Relation $S$ contains 2000 tuples and also has 10 tuples per page.
Attribute $b$ of relation $S$ is the primary key for $S$.
Both relations are stored as simple heap files.
Neither relation has any indexes built on it.
52 buffer pages are available.

(a) What is the cost of joining $R$ and $S$ using a page-oriented simple nested loops join? What is the minimum number of buffer pages required for this cost to remain unchanged?
(b) What is the cost of joining R and S using a block nested loops join? What is the minimum number of buffer pages required for this cost to remain unchanged?

(c) What is the cost of joining R and S using a sort-merge join?

3. Answer each of the questions. If some question is inapplicable, explain why. You should use the following information about R and S:

Relation R contains 200,000 tuples and has 20 tuples per page.
Relation S contains 4,000,000 tuples and also has 20 tuples per page.
Attribute a of relation R is the primary key for R.
Each tuple of R joins with exactly 20 tuples of S.
1,002 buffer pages are available.

(a) What is the cost of joining R and S using a page-oriented simple nested loops join? What is the minimum number of buffer pages required for this cost to remain unchanged?

(b) What is the cost of joining R and S using a block nested loops join? What is the minimum number of buffer pages required for this cost to remain unchanged?

(c) What would be the lowest possible I/O cost for joining Rand S using any join algorithm, and how much buffer space would be needed to achieve this cost? Explain briefly.

(d) How many tuples does the join of R and S produce, at most, and how many pages are required to store the result of the join back on disk?

4. Consider a relation with this schema:

Employees(eid: integer, ename: string, sal: integer, title: string, age: integer)

Suppose that the following indexes, all using Alternative (2) for data entries, exist: a hash index on eid, a B+ tree index on sal, a hash index on age, and a clustered B+ tree index on <age,sal>. Each Employees record is 100 bytes long, and you can assume that each index data entry is 20 bytes long. The Employees relation contains 10,000 pages, where each page contains 20 tuples.

(a) Consider each of the following selection conditions and, assuming that the reduction factor (RF) for each term that matches an index is 0.1, compute the cost of the most selective access path for retrieving all Employees tuples that satisfy the condition:

i. sal > 100
ii. age = 25
iii. age > 20
iv. eid = 1,000
v. sal > 200 ∧ age > 30
vi. sal > 200 ∧ title = ‘CFO’

(b) Suppose that, for each of the preceding selection conditions, you want to retrieve the average salary of qualifying tuples. For each selection condition, describe the least expensive evaluation method and state its cost.
(c) Suppose that, for each of the preceding selection conditions, you want to compute the average salary for each age group. For each selection condition, describe the least expensive evaluation method and state its cost.

(d) Suppose that, for each of the preceding selection conditions, you want to compute the average age for each sal level (i.e., group by sal). For each selection condition, describe the least expensive evaluation method and state its cost.

(e) For each of the following selection conditions, describe the best evaluation method:
   
i. $sal > 200 \lor age = 20$
   
ii. $sal > 200 \lor title = 'CFO'$
   
iii. $title = 'CFO' \land ename = 'Joe'$

5. You are given the following information:

Executives has attributes ename, title, dname, and address; all are string fields of the same length. The ename attribute is a candidate key.
The relation contains 10,000 pages.
There are 10 buffer pages.

Consider the following query:
SELECT E.title, E.ename FROM Executives E WHERE E.title='CFO'
Assume that only 10% of Executives tuples meet the selection condition.

(a) Suppose that a clustered B+ tree index on title is (the only index) available. What is the cost of the best plan? (In this and subsequent questions, be sure to describe the plan you have in mind.)

(b) Suppose that an unclustered B+ tree index on title is (the only index) available. What is the cost of the best plan?

(c) Suppose that a clustered B+ tree index on ename is (the only index) available. What is the cost of the best plan?

(d) Suppose that a clustered B+ tree index on address is (the only index) available. What is the cost of the best plan?

(e) Suppose that a clustered B+ tree index on $<ename, title>$ is (the only index) available. What is the cost of the best plan?

Please keep in mind:

- This is an example assignment only.
- There is no need to submit this assignment.