CMPSCI 445 — Homework 5

100 Points

Due November 24th, at the beginning of class.
Typed solutions preferred. If hand-written, solutions must be **legible**.

Query Evaluation and Optimization

1. (12 points) 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, state which of the following three approaches 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 hashed index on attribute R.a.

   (a) $\sigma_{a<50000}(R)$
   (b) $\sigma_{a=50000}(R)$
   (c) $\sigma_{a>50000 \wedge a<50010}(R)$
   (d) $\sigma_{a\neq 50000}(R)$

2. (16 points) Suppose you have a file with 1,000,000 pages and you have 13 buffer pages. Answer the following questions assuming that our most general external sorting algorithm is used.

   (a) How many runs will you produce in the first pass?
   (b) How many passes will it take to sort the file completely?
   (c) What is the total I/O cost of sorting the file?
   (d) How many buffer pages do you need to sort the file completely in just two passes?

3. (42 points) 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? What is the minimum number of buffer pages required for this cost to remain unchanged?
   (d) What is the cost of joining R and S using a hash join?
   (e) What would be the lowest possible I/O cost for joining R and S using any join algorithm, and how much buffer space would be needed to achieve this cost? Explain briefly.
   (f) 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?
   (g) If you were told that R.a is a foreign key that refers to S.b, for which of the four join algorithms above would the join costs change?

4. (30 points) 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?