

COM S 461: ASSIGNMENT 4

Percentage in your final grade: 3%

Maximum score for this assignment: 100 points

Objectives:

1. To enhance your understanding of estimation of query cost

Questions:

1. (40 points): Consider a relation with this schema:

Employee(*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 dense, unclustered B+ tree index on *sal*, a hash index on *age*, and a dense, clustered B+ tree index on (*age*, *sal*). Each Employee record is 100 bytes long, and each index data entry is 20 byte long. The Employee relation occupies 10,000 pages.

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 I/O cost of the best selective access path for retrieving all Employee tuples that satisfy the condition. Use the following assumptions in your calculation.

- Each data page can contain as many as 20 Employee tuples. Each page of Employee relation contains as many tuples as possible. The Employee relation is stored as a heap file.
 - One disk I/O is needed to retrieve a page.
 - The cost for retrieving all relevant internal nodes of a B+ tree from a root to desirable leaf node is 2 disk I/Os.
 - A page size is 2048 bytes where 48 bytes are reserved and cannot be used to store data records or data entries.
 - 1.2 I/O is needed to use hash index to find a data entry that satisfies the selection criterion.
 - As many data entries as possible are stored in a page
 - a) $sal > 100$
 - b) $age = 20$
 - c) $sal > 200$ and $age > 30$ and $title = \text{'CFO'}$
2. (60 points): Consider the join of *R* and *S* where $R.a = S.b$, given the following information about the relations to be joined. The cost metric is the number of disk I/Os and the cost of writing out the result is ignored.
 - Relation *R* contains 10,000 tuples; each tuple is 400 bytes long.
 - Relation *S* contains 2,000 tuples; each tuple is 400 bytes long.
 - The page size is 4096 bytes and the unpacked, bitmap page format is

used. For each page, 96 bytes are reserved and cannot be used to store data. The rest of the page is used to store as many tuples as possible.

- 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.
- The available memory for this join operation has 52 pages.
- The fudge factor is 1.1.

Answer the following questions and explain why:

- a) What is the cheapest cost of joining R and S using a block nested loops join for the given amount of memory space? What should the number of memory pages be so that the cost of the join is the minimum?
- b) What is the cheapest cost of joining R and S using a GRACE hash join?
- c) What is the cheapest cost of joining R and S using a sort-merge join?

Submission Requirements:

1. Put your answers in a Word document, and email it to the TA and cc to the instructor.