INFO20003 Tutorial – Week 7

(Tutorial: Query processing and cost estimation)

Objectives:

This tutorial will cover:

I. Effect of index on selection operator – 10 mins
II. Matching index – 10 mins
III. Cost estimation for different joins – 30 mins

Exercises:

1. Question about the effect of index on selection:

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 of R directly.
- Use a B+ tree index on attribute R.a.
- Use a hash index on attribute R.a.

Queries:

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

2. Matching index

Consider the following schema for the Sailors relation:

Sailors (sid INT, sname VARCHAR(50), rating INT, age DOUBLE)

For each of the following indexes, list whether the index matches the given selection conditions and briefly explain why.

- A B+ tree index on the search key (Sailors.sid)
  a. $\sigma_{\text{Sailors.sid} < 50,000}$ (Sailors)
  b. $\sigma_{\text{Sailors.sid} = 50,000}$ (Sailors)
- A hash index on the search key (Sailors.sid)
  c. $\sigma_{\text{Sailors.sid} < 50,000}$ (Sailors)
  d. $\sigma_{\text{Sailors.sid} = 50,000}$ (Sailors)
- A B+ tree index on the search key (Sailors.rating, Sailors.age)
  e. $\sigma_{\text{Sailors.rating} < 8 \wedge \text{Sailors.age} = 21}$ (Sailors)
  f. $\sigma_{\text{Sailors.rating} = 8}$ (Sailors)
  g. $\sigma_{\text{Sailors.age} = 21}$ (Sailors)

3. Question about the cost analysis of different joins:

Consider the join $R \bowtie_{R.a = S.b} S$, given the following information about the relations to be joined:

- Relation R contains 10,000 tuples and has 10 tuples/page.
- Relation S contains 2,000 tuples and also has 10 tuples/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.

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.

a. What is the cost of joining R and S using the **page-oriented Simple Nested Loops** algorithm? What is the minimum number of buffer pages (in memory) required in order for this cost to remain unchanged?

b. What is the cost of joining R and S using the **Block Nested Loops** algorithm? What is the minimum number of buffer pages required in order for this cost to remain unchanged?

c. What is the cost of joining R and S using the **Sort-Merge Join** algorithm? Assume that the external merge sort process can be completed in 2 passes.

d. What is the cost of joining R and S using the **Hash Join** algorithm?

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