CS3223 Tutorial 3: Sorting & Selection

Wk 5, Sem 2, 2022/23

Questions to be discussed: 2, 4 & 5

1. (Exercise 13.1, R&G)

   Answer the following questions (a to d) for each of the scenarios (i to iii), assuming that the external merge sort algorithm is used:

   (i) A file with 10,000 pages and three available buffer pages.

   (ii) A file with 20,000 pages and five available buffer pages.

   (iii) A file with 2,000,000 pages and 17 available buffer pages.

   (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?

2. (Exercise 13.4, R&G) Consider a disk with an average seek time of 10ms, average rotational delay of 5ms, and a transfer time of 1ms for a 4096-byte page. Assume that the cost of reading/writing a page is the sum of these values (i.e., 16ms) unless a sequence of pages is read/written. In this case, the cost is the average seek time plus the average rotational delay (to find the first page in the sequence) plus 1ms per page (to transfer data). You are given 320 buffer pages and asked to sort a file with 10,000,000 pages. Assume that you begin by creating sorted runs of 320 pages each in the first pass. Evaluate the cost of the following approaches for the subsequent merging passes:

   (i) Do 319-way merges.

   (ii) Create 256 input buffers of 1 page each, create an output buffer of 64 pages, and do 256-way merges.

   (iii) Create 16 input buffers of 16 pages each, create an output buffer of 64 pages, and do 16-way merges.

   (iv) Create eight input buffers of 32 pages each, create an output buffer of 64 pages, and do eight-way merges.

   (v) Create four input buffers of 64 pages each, create an output buffer of 64 pages, and do four-way merges.

   Complete the following table.

	Merging Factor	Input block size	Output block size	# of Merging passes	Total merging time (secs)
i	319	1	1		
ii	256	1	64		
iii	16	16	64		
iv	8	32	64		
v	4	64	64		

3. (Exercise 12.4, R&G) Consider the following schema with the Sailors relation:

   Sailors (sid: integer, sname: string, rating: integer, age: real)

   For each of the following indexes, list whether the index matches the given selection conditions. If the index is not a match, list the primary conjuncts.

   (a) A B^+-tree index on the search key `(Sailors.sid)`.
      (i) $\sigma_{Sailors.sid<50{,}000}$(Sailors)
      (ii) $\sigma_{Sailors.sid=50{,}000}$(Sailors)

   (b) A hash index on the search key `(Sailors.sid)`.
      (i) $\sigma_{Sailors.sid<50{,}000}$(Sailors)
      (ii) $\sigma_{Sailors.sid=50{,}000}$(Sailors)

   (c) A B^+-tree index on the search key `(Sailors.sid, Sailors.age)`.
      (i) $\sigma_{Sailors.sid<50{,}000 \wedge Sailors.age=21}$(Sailors)
      (ii) $\sigma_{Sailors.sid=50{,}000 \wedge Sailors.age>21}$(Sailors)
      (iii) $\sigma_{Sailors.sid=50{,}000}$(Sailors)
      (iv) $\sigma_{Sailors.age=21}$(Sailors)

   (d) A hash index on the search key `(Sailors.sid, Sailors.age)`.
      (i) $\sigma_{Sailors.sid=50{,}000 \wedge Sailors.age=21}$(Sailors)
      (ii) $\sigma_{Sailors.sid=50{,}000 \wedge Sailors.age>21}$(Sailors)
      (iii) $\sigma_{Sailors.sid=50{,}000}$(Sailors)
      (iv) $\sigma_{Sailors.age=21}$(Sailors)

4. (Exercise 12.5, R&G) Consider again the schema with the Sailors relation:

   Sailors (<u>sid</u>: integer, sname: string, rating: integer, age: real)

   where `sid` is the primary key of `Sailors`. Assume that each tuple of Sailors is 50 bytes long, that a page can hold 80 Sailors tuples, and that we have 500 pages of such tuples. For each of the following selection conditions, estimate the number of pages retrieved, given the catalog information in the question.

   (a) Assume that we have a B^+-tree index T on the search key `(Sailors.sid)`, and assume the following:
      - the number of internal (ie non-leaf) levels in $T = 4$,
      - the number of leaf pages in $T = 50$, and
      - $Sailors.sid \in [1, 100, 000]$

      (i) $\sigma_{Sailors.sid<50{,}000}$(Sailors)
      (ii) $\sigma_{Sailors.sid=50{,}000}$(Sailors)

   (b) Assume that we have a hash index T on the search key `(Sailors.sid)`, and assume the following:
      - the number of primary and overflow pages in $T = 50$, and
      - $Sailors.sid \in [1, 100, 000]$

      (i) $\sigma_{Sailors.sid<50{,}000}$(Sailors)
      (ii) $\sigma_{Sailors.sid=50{,}000}$(Sailors)

5. Consider a relation with the schema `Hotel` (id, name, address, phone, price, rating) and the following query

   SELECT name FROM Hotel WHERE price > 500 AND rating > 4

   Assume the following for this question:

   1. The `Hotel` relation contains 10,000 pages with each page containing 20 records.
   2. There are five unclustered indexes on the `Hotel` relation:
      - I_p: a B^+-tree index on (price),
      - I_r: a B^+-tree index on (rating),
      - I_{pr}: a B^+-tree index on (price, rating),
      - I_{rp}: a B^+-tree index on (rating, price), and
      - I_{npr}: a B^+-tree index on (name, price, rating).
   3. For each of I_p and I_r, it has 2000 leaf pages with at most 100 data entries in each leaf page.
   4. For each of I_{pr} and I_{rp}, it has 2500 leaf pages with at most 80 data entries in each leaf page.
   5. For I_{npr}, it has 4000 leaf pages with at most 50 data entries in each leaf page.
   6. Each of the indexes has 2 levels of internal nodes.
   7. Only 10% of `Hotel` records satisfy the condition "price > 500".
   8. Only 20% of `Hotel` records satisfy the condition "rating > 4".
   9. Only 1% of `Hotel` records satisfy both the conditions "price > 500" and "rating > 4".

   What is the best evaluation plan for the query? What is its cost?