Section 2: The actual mid-term exam

2. Consider a buffer pool that can hold 3 pages only, and the following sequence of requests, where each alphabet is the ID of a disk page.

A B C C A D B E A D A

Suppose we use the LRU ("least recently used") replacement policy.

(1 mark)

Option 1: The number of disk I/Os is 8.
Option 2: The pages in the buffer at the end are: A D A
Option 3: The number of hits (i.e., the number of times a wanted page is already in the buffer) is 2.
Options 1 and 2 only
Options 2 and 3 only
Options 1, 2 and 3
None of the options

3. Consider a hard disk with the following specifications.

- 3.5in in diameter
- 3840 RPM
- 10 platters, and 2 surfaces each platter
- Usable capacity: 10GB (2^30 B)
- Number of cylinders: 256
- 1 block = 4 KB (2^12 B)
- 20% overhead between blocks (gaps)
- Average seek time: 20 ms.

What is the number of blocks per track?

(1 mark)

- 64
- 128
- 256
- 512

4. Referring to the previous question, the *burst bandwidth* of reading a single block can be computed by dividing the size of a block (4KB in our context) by the time to read one block *without a gap.* What is the burst bandwidth for our device.

(1 mark)

- 40 MB/s
- 80 MB/s
- 120 MB/s
- 160 MB/s
- None of the above

5. Consider the following sequence of numbers:

22, 44, 7, 39, 49, 12, 89, 10, 23, 55, 50, 67, 33, 28, 11, 40, 46

Suppose we use replacement selection to generate sorted runs. Moreover, suppose each page has only 1 record. Let the number of buffer pages be 3. What is the length of the shortest run?

(1 mark)

4

5

6

7

None of the above

6. Referring to the previous question, what is the number of sorted runs generated?

(1 mark)

3

4

5

None of the above

7. Which of the following statements is true?

(1 mark)

Given a two level (root plus leaf level - leaf level contains pointer to data records) B+ tree of order 2. The minimum number of record pointers at the leaf is 5.

The B+-tree is order independent, i.e., regardless of the order of input, the resultant B+-tree will always have the same structure.

Every separator in the internal nodes of a B+-tree must appear in the leaf node.

All of the above 3 options.

None of the above options.

8. Insert the following sets of keys to an initially empty B+-tree of order 1:

5, 20, 50, 30, 26, 60, 40, 35

Which of the following statements is/are true?

(1 mark)

Option1: The height of the tree is 3.

Option 2: The value 35 appears multiple times in the tree.

Option 3: In total, 4 nodes are full.

Only options 1 and 2

Only options 2 and 3

Only options 1 and 3.

Options 1, 2 and 3 are all correct.

9. Referring to the previous question on B+-tree. Now, delete 5. What is the resultant structure like?

(1 mark)

- Option 1: The height of the tree is 2.
- Option 2: In total, there are 3 full nodes.
- Option 3: Some key values in the internal nodes no longer appear in the leaf nodes.
- Only options 1 and 2.
- Only options 2 and 3.
- Only options 1 and 3.
- None of the above.

10. Consider the join between R and S. Let |R| = 1000, and |S| = 1000. Let the number of buffers be 200. Assume that R and S are uniformly distributed. Suppose a random I/O costs 15ms and a sequential I/O costs 1ms. Consider the GRACE hash join. we shall consider the partitioning phase in this question. Suppose we use 100 buffers for input, and consider the following method (referred to as Method A) to partition R (S is partitioned in a similar way).

A) We generate 100 partitions, i.e., allocate one (1) output buffer per partition.

What is the partitioning cost of R?

- Cost of method A = 2000
- Cost of method A = 30000
- Cost of method A = 2140
- Cost of method A = 16140

None of the above.

11. Referring to the previous question. What is the final join cost? We assume that the output is not written to disks (i.e., the output tuples are returned to the users once they are generated).

- 6000
- 90000
- 62280
- 20460
- 34680
- 37080
- None of the above.

12. Referring to the previous question. Besides Method A (reproduced here), let's consider also Method B:

A) We generate 100 partitions, i.e., allocate one (1) output buffer per partition.

B) We generate 10 partitions, i.e., allocate ten (10) output buffers per partition.

Which of the two methods result in a lower GRACE hash join cost?

- Method A
- Method B

Both have the same cost

Not enough information to make the comparison.

13. Which of the following statements is/are true?

(1 mark)

- Option 1: In Linear Hashing, the order in which records are inserted may lead to different hash structure.
- Option 2: Linear hashing extends by one bucket at a time (instead of multiple buckets each time) because allocating more buckets will lead to more overflows.
- Option 3: In linear hashing, one possible splitting criterion is: split whenever the bucket pointed to by the next pointer overflows. This criterion is preferred over the default (split whenever there is an overflow bucket) because it can clear long overflow chains quickly.
- Options 1 and 3 only.
- Options 2 and 3 only.
- None of the above.

14. Consider an extendible hash index with directory size of 16. Which of the following statements is/are true?

(1 mark)

- We have at most 16 data buckets.
- It is possible to have 2 data buckets - each with 8 directory pointers pointing to it.

It is possible to have 2 data buckets - one has one directory pointer pointing to it, and the other has the other 15 directory pointers pointing to it.

It is possible to have 2 data buckets - one is full, and the other is empty.

Options 1 and 2.

Options 2 and 3.

Options 1 and 3.

None of the above.

15. Consider an extendible hash table. Which of the following statements is true?

(1 mark)

Option 1: The local depth can be smaller than the global depth.

Option 2: Whenever there is an insertion, and the local depth of a bucket is equal to the global depth, the directory size doubles.

Option 3: Whenever there is an insertion, and the local depth of a bucket is equal to the global depth, redistribution of data is necessary.

Option 4: Whenever an insertion results in the doubling of directory, there are exactly two buckets each of which has one directory pointers pointing to it.

Options 1 and 3 are correct.

Options 1 and 4 are correct.

Options 2 and 3 are correct.

Options 1, 2 and 4

16. Consider relation R(a, b, c) where all attributes are of the same size. Suppose 10 records fit in a page, and the table has 1000 records. Suppose we have 10 pages of buffer. Given the query: SELECT distinct R.a FROM R. Suppose we use the optimized sort-based scheme for duplicate elimination. How many sorted runs are generated?

(1 mark)

- 2
- 4
- 8
- 30
- 100

17. Referring to the previous question, suppose 50% of R.a are duplicates. What is the total number of I/Os to complete the duplication elimination (including the cost to write the output).

(1 mark)

- 400
- 350
- 218
- 185
- Not enough information to compute.

18. Consider the join between R and S, say on join condition R.a = S.b. We are given the following information:

R contains 10,000 tuples and has 10 tuples per page.
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.

What is the minimum I/O 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?

(1 mark)

- cost = 4200; buffer = 50
- Cost = 4200; buffer = 52
- Cost = 1200; buffer = 202
- None of the above.

19. Referring to the previous join question. What is the maximum join output size (in tuples), and the maximum number of output pages? We assume each output tuple contains all the attributes.

(1 mark)

- Not enough information to compute.
- Number of tuples = 2000; Number of pages = 200
- Number of tuples = 10000; Number of pages = 1000

Number of tuples = 10000; Number of pages = 2000

Number of tuples = 2000; Number of pages = 400

20. Consider a heap file containing 20,000 records. Assume a dense B+ tree index is built using format 2 (i.e., (key,pointer)-pairs at the leaf levels). The key field for this B+ tree index is a 40-byte string, and it is a candidate key. Pointers are 10-byte values. The size of one disk page is 1000 bytes. We can assume that the nodes at each level were filled up as much as possible (i.e., minimum number of nodes are needed).

(1 mark)

Option 1: The order of the tree is 9.

Option 2: The height of the tree is 4.

Option 3: There are 2 nodes at the level below the root.

Options 1 and 2.

Options 2 and 3.

Options 1 and 3.

Options 1, 2 and 3.

Save For Later

Finish Quiz