Assignment 2

Due Date: April 18, 2025, at 11:59 pm

10% of the final grade

NAME: ____________________

UPI: ____________________

ID: ____________________

Please ensure that you fill in your name, UPI, and ID above.

It's best to answer Q2 after completing your Week 5 lab, and Q3 after completing your Week 6 lab.

Completely fill the circles as shown: ○○●○

Q1 [**3 marks**] Assume that each block is 1KB, and the buffer pool is 1MB.

1. [**1 mark**] What is the maximum file size that can be sorted using external memory sorting in two passes? Recall that the first pass follows the create-runs step.

   ● a. 1TB
   ○ b. 10GB
   ○ c. 1GB
   ○ d. 100MB
   ○ e. None of the above

   Marking Rubric. The recommended method is to estimate the largest file size sorted using two passes and then select the closest matching option. Based on this approach, option a is the most appropriate answer. While other interpretations may lead to options like b or e (which we give full marks), the question is primarily concerned with identifying the approximate scale of the file size.

2. [**1 mark**] Consider a 10GB relation and construct a B+ tree for the relation using external memory sorting. Assume that all leaf nodes (each stored in a block) are full and that all internal nodes reside in the main memory. What is the total number of I/Os required to construct the B+ tree? Note:

   i. The leaf nodes of the B+ tree are **stored** on disk.
   ii. Constructing the first-level index of the B+ tree requires reading the sorted file of the relation once.

   ● a. 70M
   ○ b. 80M
   ○ c. 90M
   ○ d. 100M
   ○ e. None of the above

   The formula does not account for write-to-disk I/Os of the last pass. In our case, all tuples must be written to disk as part of the downstream operation, and then read again during index construction.

3. [**1 mark**] A colleague is preparing to interview a candidate and would like to assess their knowledge of indexing. The colleague has compiled a list of statements and requests that you identify which ones are true. Please select all the correct statements.

   ● Write-optimized indices can significantly reduce the cost of inserts, and to a lesser extent, of updates, as compared to B+trees. On the other hand, the index lookup cost can be significantly higher for write-optimized indices as compared to B+trees.
   ○ NULL values can be easily treated because they represent the absence of a value, making it straightforward to handle them without requiring special attention.
   ● Bloom filters can eliminate unnecessary disk I/Os.
   ○ Bloom filters are effective for exact-match (or lookup) queries.
   ○ None of the above

   Marking Rubric: Selecting options 1, 3, and 4 also earns full marks.

Q2 [**4 marks**] Consider the B^+-tree shown in Figure 1 with two levels of nodes. Each leaf node occupies a block, as does each internal node. Each block can hold up to 3 tuples, or alternatively, it can function as an internal node with a fanout of 5. Answer the following questions.

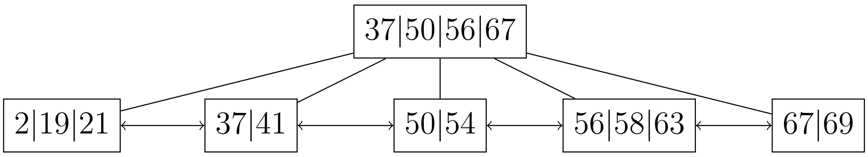
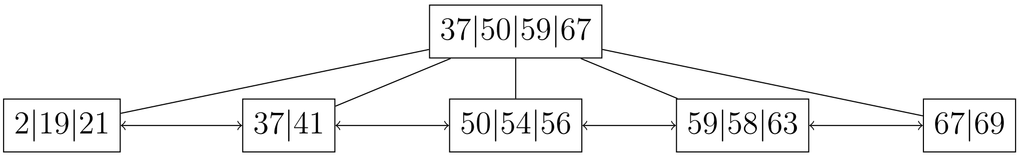


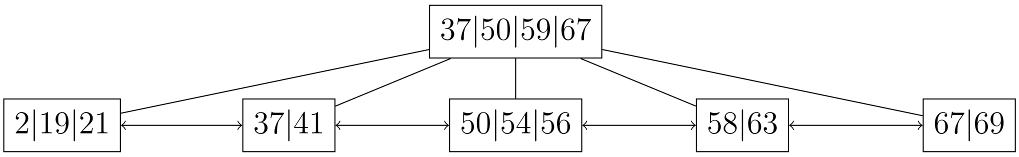
Figure 1: B^+-tree

1. [**1 mark**] Insert 59 to the B^+ tree of Figure 1. Select the resulting tree.

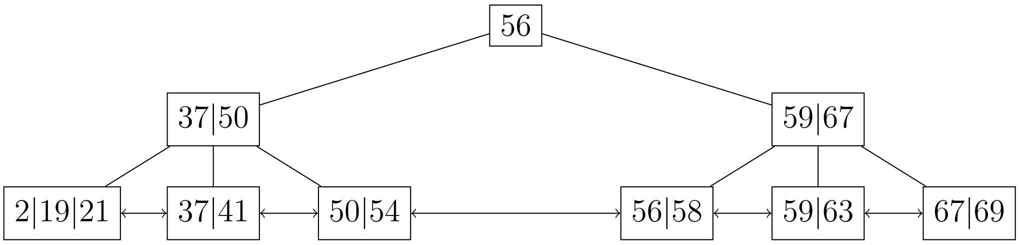
○ a.



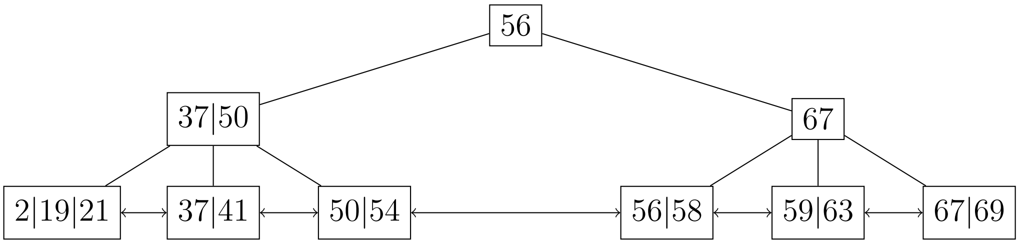
○ b.



● c.



○ d.



○ e. None of the above

2. [**1 mark**] After inserting 59 to Figure 1, how many I/Os are need to find all the tuples with keys in an open range of $(35, 57)$? Assume the root is stored in the main memory and exclude the I/O cost of reporting the output.

○ a. 2
○ b. 3
○ c. 4
● d. 5
○ e. None of the above

3. [**1 mark**] During the sequential insertion of the values 59, 4, 8, 1, 5, and 7 into the B+ Tree shown in Figure 1, how many nodes were split?

○ a. 2
○ b. 3
○ c. 4
○ d. 5
● e. None of the above

4. [**1 mark**] How many tuples a B^+tree of 4 levels can store at most under current parameters?

○ a. 75
● b. 375
○ c. 1875
○ d. 150
○ e. None of the above

Q3 [**3 marks**] Consider an extendible hashing structure such that:

- Each bucket can hold up to three records.
- The hashing function uses the highest g bits (left bits are high) of the hashing value, where g is the global depth.
- A new extendible hashing structure is initialized with g = 0 and one empty bucket.
- If multiple keys are provided in a question, assume they are inserted one after the other from left to right.
- Records with duplicate keys will be retained without deduplication.

Key	Hashing value	Key	Hashing value
7	000	18	100
20	110	25	100
37	010	44	010
49	000	50	001
51	010	69	110

1. [**1 mark**] After inserting 50, 44, 25, 20, 37, what are the local depth of the bucket containing 25 and the global depth, respectively?

○ a. 25 (depth 0). The global depth is 1.
● b. 25 (depth 1). The global depth is 1.
○ c. 25 (depth 2). The global depth is 1.
○ d. 25 (depth 1). The global depth is 2.
○ e. None of the above

2. [**1 mark**] Starting from the results of question Q3.1, insert 7. What are the local depths of the buckets for each key?

● a. 7 (depth 2), 20 (depth 1), 25 (depth 1), 37 (depth 2).
○ b. 7 (depth 2), 20 (depth 2), 25 (depth 1), 37 (depth 2).
○ c. 7 (depth 2), 20 (depth 1), 25 (depth 2), 37 (depth 2).
○ d. 7 (depth 1), 20 (depth 1), 25 (depth 1), 37 (depth 2).
○ e. None of the above

3. [**1 mark**] Starting from the results of question Q3.2, insert 51, 49, 18, 69, 37. What are the local depths of the buckets for each key? What is the global depth?

○ a. 20 (depth 2), 25 (depth 2), 37 (depth 3), 50 (depth 2). Global depth: 3.
○ b. 20 (depth 2), 25 (depth 2), 37 (depth 3), 50 (depth 3). Global depth: 3.
● c. 20 (depth 2), 25 (depth 2), 37 (depth 2), 50 (depth 2). Global depth: 2.
○ d. 20 (depth 2), 25 (depth 1), 37 (depth 3), 50 (depth 2). Global depth: 3.
○ e. None of the above