**CS 5600 Advanced Database Systems**

**Midterm Exam Review**

1. (a) Suppose that the following is a leaf node of a B+ tree. Which pointer is the address of the data block containing a search key of value 9?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| P1 | 5 | P2 | 9 | P3 | 15 | P4 |

1. Suppose that the following is an internal node of a B+ tree. Which pointer is the address of the child node for a search key value 14?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| P1 | 5 | P2 | 9 | P3 | 15 | P4 |

1. Insert 20 into the following B+-tree.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 10 |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 30 |  | 40 | \* | 50 | \* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| \* | 5 | \* |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| \* | 10 | \* | 15 | \* | 18 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| \* | 30 | \* | 35 |  |  |  |

1. Delete 10 from the following B+-tree.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 10 |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 30 |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| \* | 5 | \* |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| \* | 10 | \* | 20 |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| \* | 30 | \* | 35 |  |  |  |

1. Assume the hash function is *h(x) = x mod 8* and buckets can hold three records.



1. Show the changes made to the given extendable hash structure as the result of first inserting 34 and then deleting 17. Please draw the new diagram.
2. Show the changes made to the resulting extendable hash structure from Step a. as the result of inserting 5 and then inserting 39. Please draw the new diagram.
3. Given the following relation with attributes A, B, C, and D.

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | D |
| 111 | Ccc | A1 | 10000 |
| 222 | Aaa | B2 | 20000 |
| 333 | Bbb | D4 | 40000 |
| 444 | Eee | C3 | 60000 |
| 555 | Hhh | A1 | 30000 |
| 666 | Ggg | C3 | 50000 |
| 777 | Iii | B2 | 40000 |
| 888 | Ddd | D4 | 20000 |
| 999 | Fff | A1 | 50000 |

1. Construt a bitmap index on the attribute D, dividing the values of D into 3 ranges: R1 = (0, 35000]; R2 = (35000, 55000]; R3 = (55000, 90000].
2. Consider a query that request all values of B with the value of C equals to “A1” and the values of D greater than (>) 25000. Show the final and intermediate bitmaps constructed to answer the query.

6.

1. Assume that the block size is 4k and it takes 4 bytes to store a key value and 6 bytes to store a pointer in B+-tree nodes. What is the number of pointers in the tree node?
2. What is the main difference between a primary and a secondary index?
3. Assume that there are 16 blocks available in memory for external sort-merge. How many merge-passes are there to sort a relation with 512 blocks?
4. Consider a selection query of equality on a key attribute. Assume that a primay B+-tree index on the key attribute is used in the search, and the tree has 100 levels. How many seeks and block transfers are needed?
5. Consider a selection query of equality on a nonkey attribute. Assume that a secondary B+-tree index on the nonkey attribute is used in the search, and the tree has 100 levels. Assume also that there are 15 blocks containing records with the specified search key. How many seeks and block transfers are needed?
6. Suppose that relations r and s contain 20 and 30 blocks respectively, and the block nested-loop join algorithm is used to natural-join r and s. What is the cost in terms of number of seeks and number of block transfers in a best case senarious? How many blocks of memory are required to achieve this cost in the best case senarous?
7. Let relations *r*1(*A*, *B*,*C*) and *r*2(*C*, *D*, *E*) have the following properties: *r*1 has 50,000 tuples, *r*2 has 40,000 tuples, 20 tuples of *r*1 fit on one block, and 50 tuples of *r*2 fit on one block. Estimate the number of block transfers and seeks required, using each of the following join strategies for *r*1 natural join *r*2:
8. Nested-loop join.

b. Block nested-loop join.

1. Beable to derive the number of seeks and number of block transfers for algoriths A1 – A6 in chap 12.