Name

CWID

Quiz
1

Oct 10, 2018
Due Oct 14, 11:59pm

Quiz 1: CS525 - Advanced Database Organization

Results

Please leave this empty! 1.1 1.2 1.3 Sum

Instructions

- **You have to hand in the assignment using your blackboard**
- **This is an individual and not a group assignment. Fraud will result in 0 points**
- For your convenience the number of points for each part and questions are shown in parenthesis.
- There are **3** parts in this quiz
  1. Disk Organization
  2. SQL
  3. Index Structures

BY SUBMITTING THIS EXAM THROUGH THE ONLINE SYSTEM, I AFFIRM ON MY HONOR THAT I AM AWARE OF THE STUDENT DISCIPLINARY CODE, AND (I) HAVE NOT GIVEN NOR RECEIVED ANY UNAUTHORIZED AID TO/FROM ANY PERSON OR PERSONS, AND (II) HAVE NOT USED ANY UNAUTHORIZED MATERIALS IN COMPLETING MY ANSWERS TO THIS TAKE-HOME EXAMINATION.

Part 1.1 Disk Organization (Total: 20 Points)

Question 1.1.1 Disk Access (20 Points)

Consider a disk with a sector size of 512 bytes, 2000 tracks per surface, 50 sectors per track, five double-sided platters, and average seek time of 10 msec. Suppose that a block size of 1024 bytes is chosen. Suppose that a file containing $100,000$ records of 100 bytes each is to be stored on such a disk and that no record is allowed to span two blocks.

1. How many records fit onto a block?

   Solution

   (3 Points) $\frac{block\ \ size}{record\ \ size} = \frac{1024}{100} = 10$.

   $\Longrightarrow$ `We can have at most 10 records in a block.`

2. How many blocks are required to store the entire file?

   Solution

   (3 Points) `There are 100,000 records all together, and each block holds 10 records. Thus, we need 10,000 blocks to store the file.`

3. If the file is arranged sequentially on the disk, how many surfaces are needed?

   Solution

   (3 Points) $bytes/track = bytes/sector \times sectors/track = 512 \times 50 = 25Kbytes$ $\Longrightarrow$ `One track has 25 blocks, one cylinder has` $25 \times 5 \times 2 = 250$ `blocks.`
   `So, we need 10,000 blocks to store this file.`
   `So we will use more than one cylinders, that is, need` **10 surfaces** `to store the file.`

4. How many records of 100 bytes each can be stored using this disk?

   Solution

   (3 Points) `The capacity of the disk is 500,000K, which has 500,000 blocks. Each block has 10 records. Therefore, the disk can store no more than` **5,000,000 records**.

5. If pages are stored sequentially on disk, with page 1 on block 1 of track 1, what page is stored on block 1 of track 1 on the next disk surface?

   Solution

   (4 Points) `There are 25 blocks in each track, or we can say, 25 blocks in each track. It is block 26 on block 1 of track 1 on the next disk surface.`

6. How would your answer change if the disk were capable of reading and writing from all heads in parallel?

   Solution

   (4 Points) `If the disk were capable of reading/writing from all heads in parallel, we can put the first 10 pages on the block 1 of track 1 of all 10 surfaces. Therefore, it is block 2 on block 1 of track 1 on the next disk surface.`

Part 1.2 SQL (Total: 35 Points)

Consider the following relations:

- `Suppliers`(<u>`sid`:**integer**</u>, `sname`:**string**, `address`:**string**)
- `Parts`(<u>`pid`:**integer**</u>, `pname`:**string**, `color`:**string**)
- `Catalog`(<u>`sid`:**integer**, `pid`:**integer**</u>, `cost`:**real**)

The key fields are underlined, and the domain of each field is listed after the field name. Therefore `sid` is the key for `Suppliers`, `pid` is the key for `Parts`, and `sid` and `pid` together form the key for `Catalog`. The `Catalog` relation lists the prices charged for parts by `Suppliers`. Write the following queries in SQL statements and in relational algebra expression.

Question 1.2.1 (5 Points)

Find the `names` of suppliers who supply some red part

Solution

```
SELECT S.sname
FROM Suppliers S, Parts P, Catalog C
WHERE P.color='red' AND C.pid=P.pid AND C.sid=S.sid
```

$$\pi_{name}\left(\pi_{sid}(\pi_{pid}(\sigma_{color='red'}Parts) \bowtie Catalog) \bowtie Suppliers\right)$$

Question 1.2.2 (5 Points)

Find the `sids` of suppliers who supply some red or green part.

Solution

```
SELECT C.sid
FROM Catalog C, Parts P
WHERE (P.color='red' OR P.color='green') AND P.pid = C.pid
```

$$\pi_{sid}(\pi_{pid}(\sigma_{color='red' \vee color='green'}Parts) \bowtie Catalog)$$

Question 1.2.3 (5 Points)

Find the sids of suppliers who supply some red part or are at 10 West 31st Street.

Solution

```
SELECT S.sid
FROM Suppliers S
WHERE S.address = '10th West Street'
      OR S.sid IN ( SELECT C.sid
                    FROM Parts P, Catalog C
                    WHERE P.color='red' AND P.pid = C.pid )
```

$$\pi_{sid}(\pi_{pid}(\sigma_{color='red'}Parts) \bowtie Catalog) \cup \pi_{sid}(\sigma_{address='10\ West\ 31st\ Street'}Suppliers)$$

Question 1.2.4 (5 Points)

Find the sids of suppliers who supply some red part and some green part.

Solution

```
SELECT C.sid
FROM Parts P, Catalog C
WHERE P.color = 'red' AND P.pid = C.pid
   AND EXISTS (SELECT P2.pid
          FROM Parts P2, Catalog C2
          WHERE P2.color = 'green' AND C2.sid = C.sid AND P2.pid = C2.pid)
```

$$\pi_{sid}\Big(\big(\pi_{pid}(\sigma_{color='red'}Parts) \cap \pi_{pid}(\sigma_{color='green'}Parts)\big) \bowtie Catalog\Big)$$

Question 1.2.5 (5 Points)

Find pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second sid.

Solution

```
SELECT R1.sid, R2.sid
FROM Catalog R1, Catalog R2
WHERE R1.pid = R2.pid AND R1.sid ≠ R2.sid AND R1.cost > R2.cost
```

$$\begin{aligned} &\rho_{R_1}Catalog \\ &\rho_{R_2}Catalog \\ E \leftarrow \rho_{R_1}Catalog \times &\rho_{R_2}Catalog \end{aligned}$$

$$\pi_{R_1.sid,R_2.sid}\big(\sigma_{R_1.pid=R_2.pid \wedge R_1.sid \neq R_2.sid \wedge R_1.cost>R_2.cost}(E)\big)$$

Question 1.2.6 (5 Points)

Write only an SQL query that find the pids of parts supplied by at least two different suppliers.

Solution

```
SELECT R1.pid
FROM Catalog R1
WHERE EXISTS (SELECT R2.sid
              FROM Catalog R2
              WHERE R2.pid = R1.pid AND NOT R2.sid = R1.sid )
```

Question 1.2.7 (5 Points)

Write only an SQL query that find the pids of the most expensive parts supplied by suppliers named Yosemite Sham.

Solution

```
SELECT C.pid
FROM Catalog C, Suppliers S
WHERE S.sname = 'Yosemite Sham' AND C.sid = S.sid
         AND C.cost ≥ ALL (Select C2.cost
                           FROM Catalog C2, Suppliers S2
                           WHERE S2.sname = 'Yosemite Sham'
                           AND C2.sid = S2.sid)
```

Part 1.3 Index Structures (Total: 65 Points)

Question 1.3.1 B^+-tree Construction (15 Points)

Assume that you have the following table:

Item

SSN	name	age
2	Pete	13
23	Bob	23
5	John	49
39	Joe	45
29	Alice	77
17	Lily	3
19	Manny	33
7	Gertrud	29
11	Heinz	14
3	Sammy	34

Create a `B+-tree` for table **Item** on key *SSN*. Assume that the tree is initially empty and values are added in ascending order. Construct `B+-tree` for the cases where the number of pointers that will fit in one node is as follows:

a. Three

b. Five

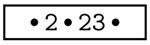
Write down the resulting `B+-tree` after each step and when splitting or merging nodes follow these conventions:

- **Leaf Split**: In case a leaf node needs to be split during insertion and n is even, the left node should get the extra key. E.g, if $n = 2$ and we insert a key 4 into a node [1,5], then the resulting nodes should be [1,4] and [5]. For odd values of n we can always evenly split the keys between the two nodes. In both cases the value inserted into the parent is the smallest value of the right node.
- **Non-Leaf Split**: In case a non-leaf node needs to be split and n is odd, we cannot split the node evenly (one of the new nodes will have one more key). In this case the "middle" value inserted into the parent should be taken from the right node. E.g., if $n = 3$ and we have to split a non-leaf node [1,3,4,5], the resulting nodes would be [1,3] and [5]. The value inserted into the parent would be 4.
- **Node Underflow**: In case of a node underflow you should first try to redistribute values from a sibling and only if this fails merge the node with one of its siblings. Both approaches should prefer the left sibling. E.g., if we can borrow values from both the left and right sibling, you should borrow from the left one.

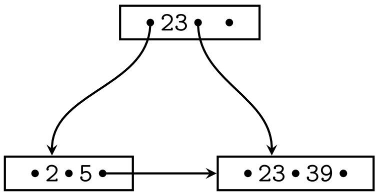
Solution

a. Three

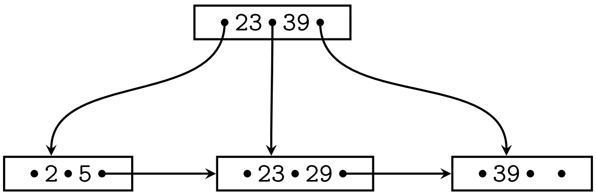
Insert 2, 23



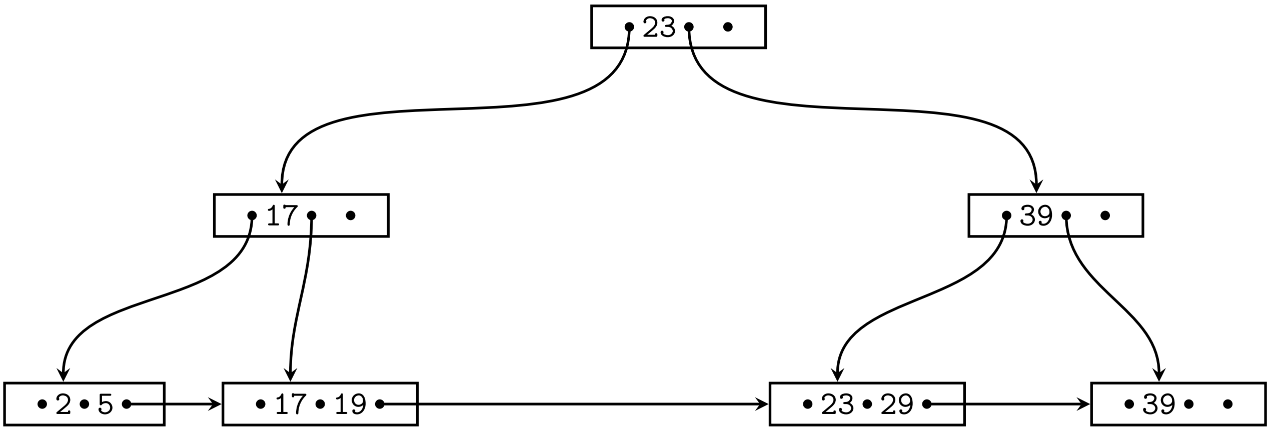
Insert 5, 39



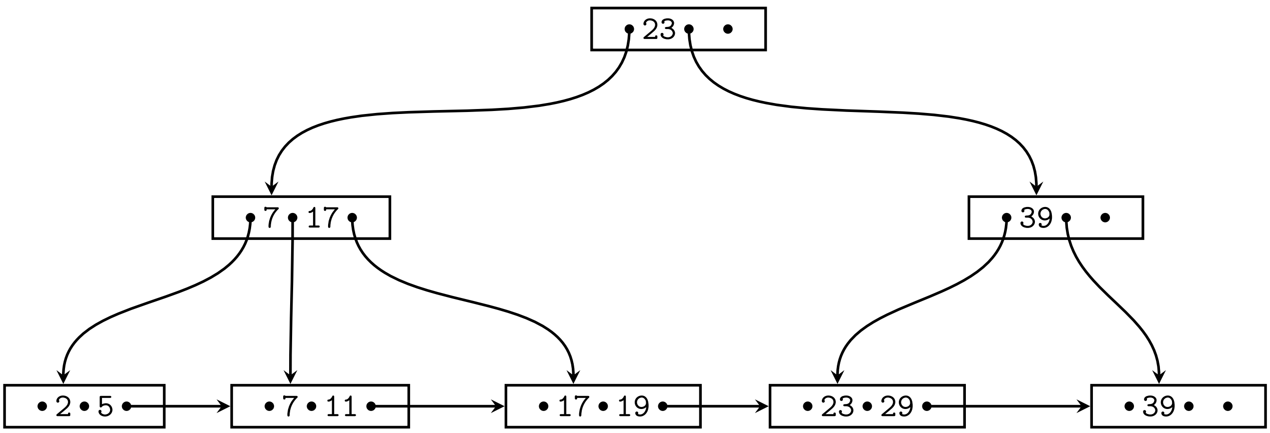
Insert 29



Insert 17, 19

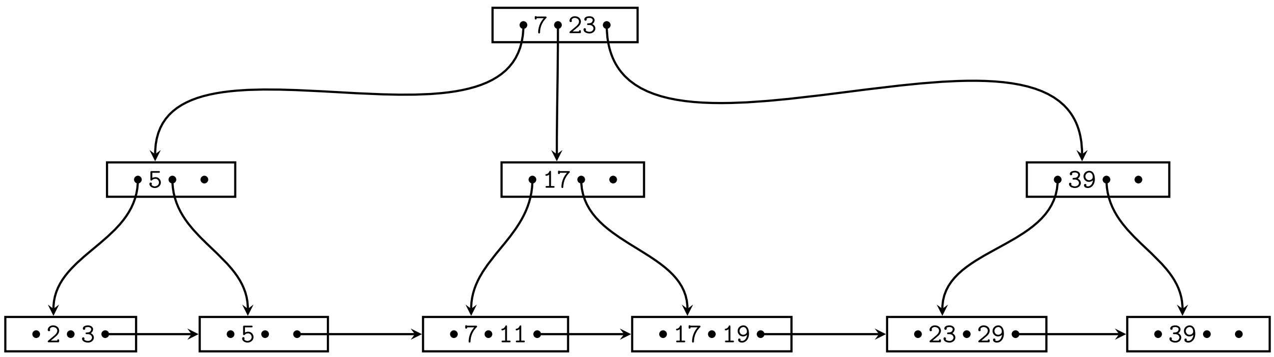


Insert 7, 11



Solution

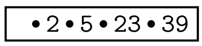
Insert 3



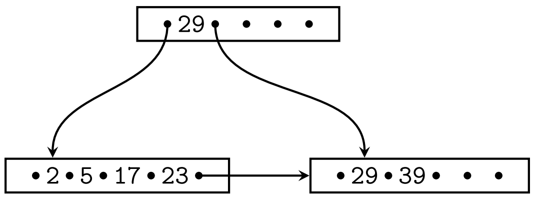
Solution

b. Five

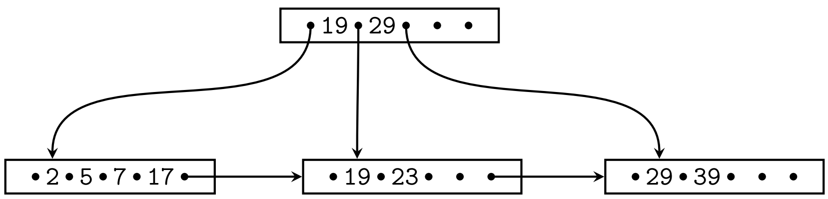
Insert 2, 23,5,39



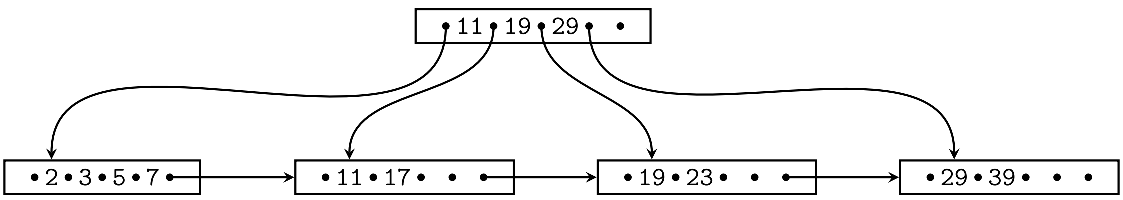
Insert 29, 17



Insert 19, 7



Insert 11, 3



Question 1.3.2 Extensible Hashing-tree Construction (25 Points)

Suppose that we are using `Extensible Hashing` on a file that contains records with the following search key values: `3, 11, 7, 19, 14, 18, 15, 17, 20, 44, 33, 43`

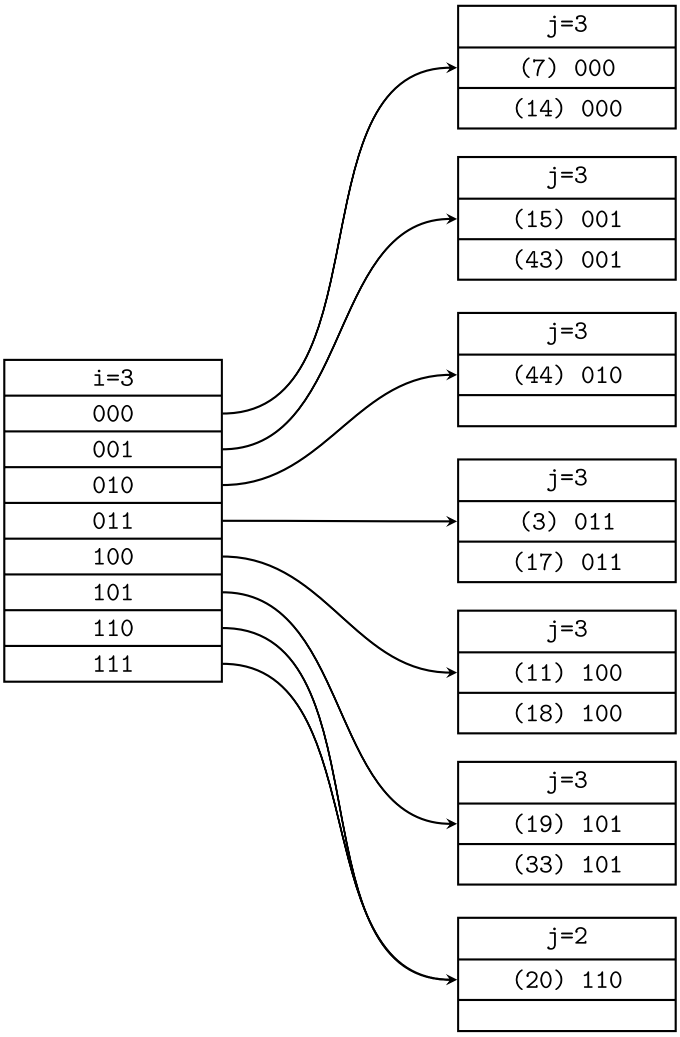
Show the `Extensible Hash` structure for this file if the hash function is `h(x)=x` mod 7 and buckets can hold

a. 2 records

b. 3 records

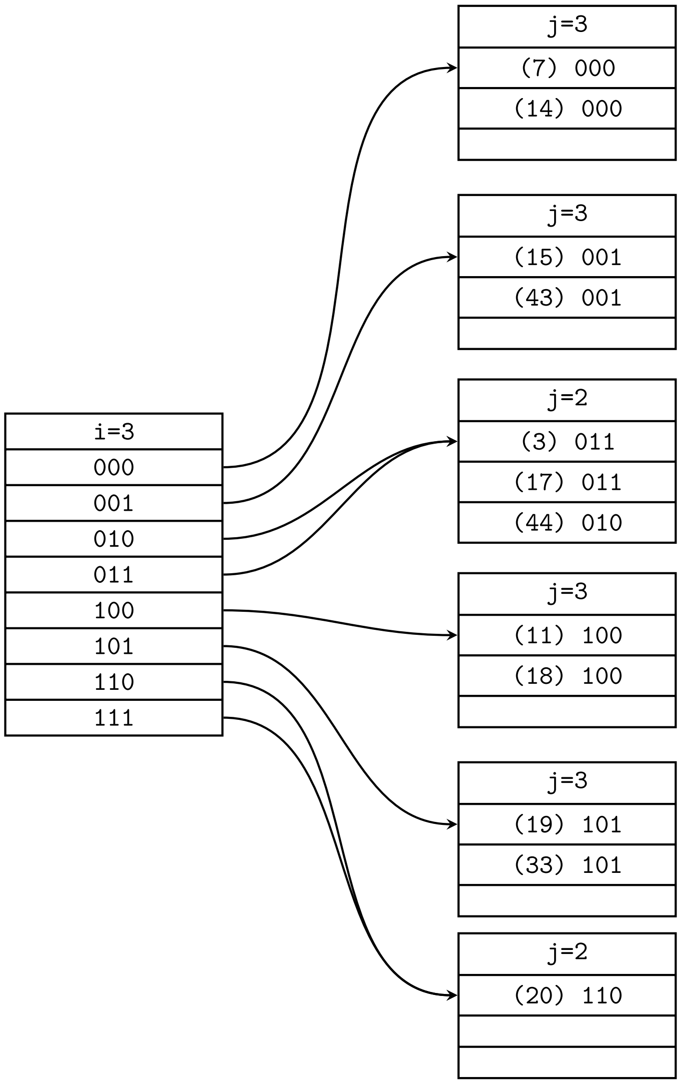
Solution

a) Final Result: You need to show your work



Solution

b) Final Result: You need to show your work

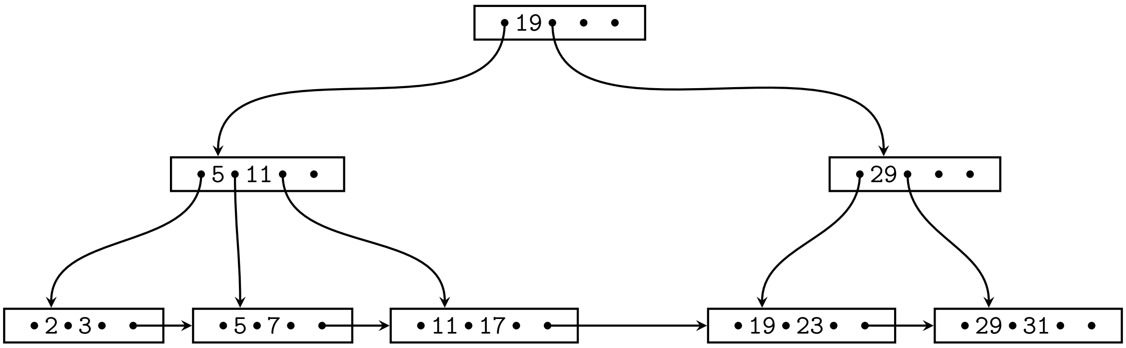


Question 1.3.3 Operations (25 Points)

Given is the `B+-tree` shown below ($n = 3$). Execute the following operations and write down the resulting `B+-tree` after each operation:

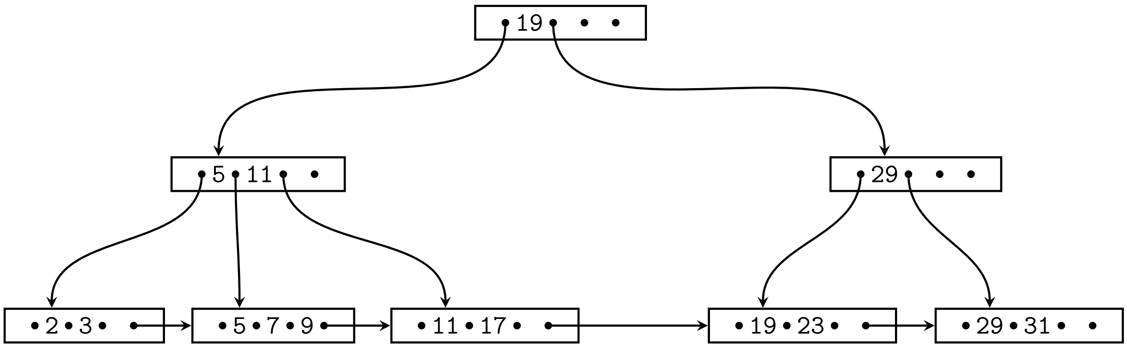
insert(9), delete(23), insert(10), insert(8), delete(19), delete(11)

Use the conventions for splitting and merging introduced in the previous question.

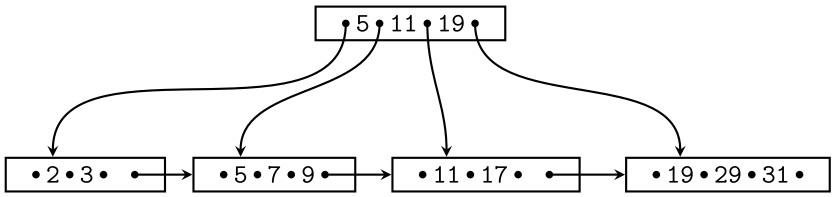


Solution

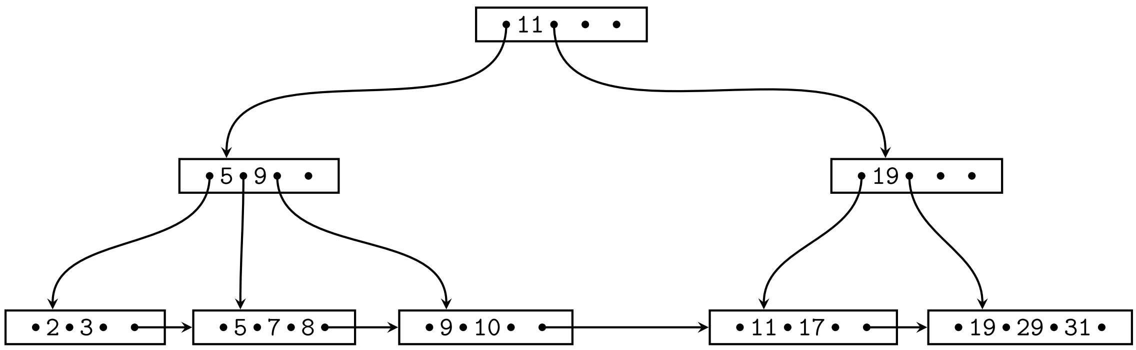
insert(9)



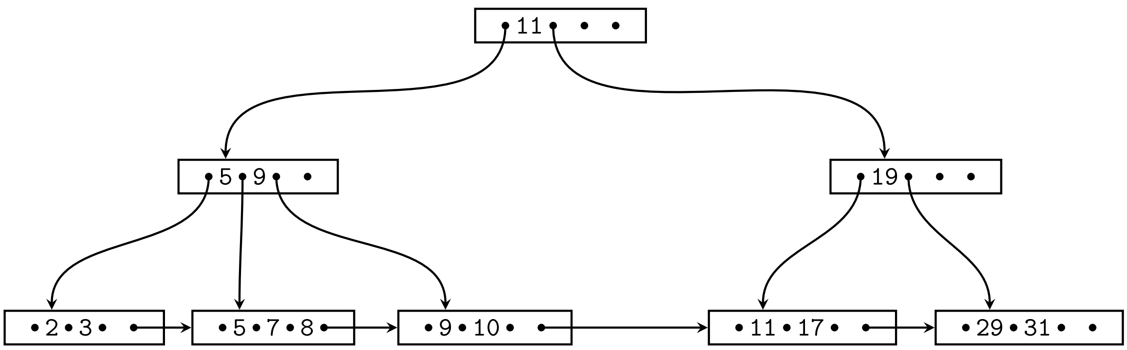
delete(23)



insert(10), insert(8)



delete(19)



delete(11)