# CSCI 6333/6315 Database Systems

# Spring 2020

## ASSIGNMENT 4: Data Storage and Querying, Transaction Management

## Sample Solutions

Problem 1. This problem has two parts:

a. Construct a $B^+$-tree for the following key values:

2, 3, 5, 7, 11, 17, 19, 23, 29, 31.

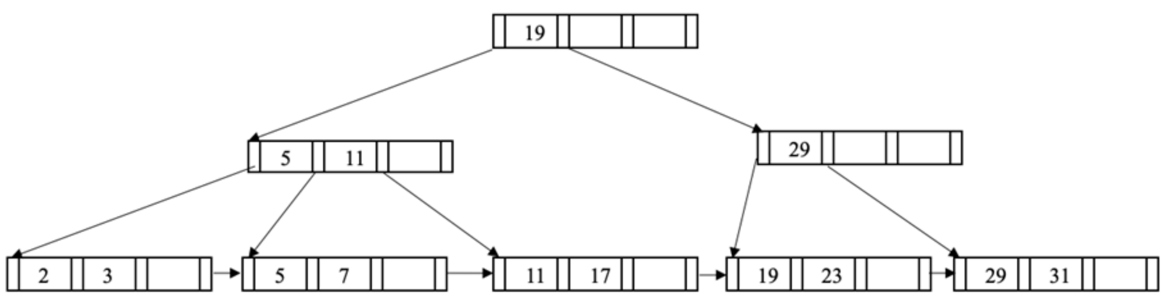
Assume that the number of pointers that will fit in one internal node is 4 and each leaf node can store 3 key values.

b. After the $B^+$-tree is constructed for Part a, show the final tree after the following operations:

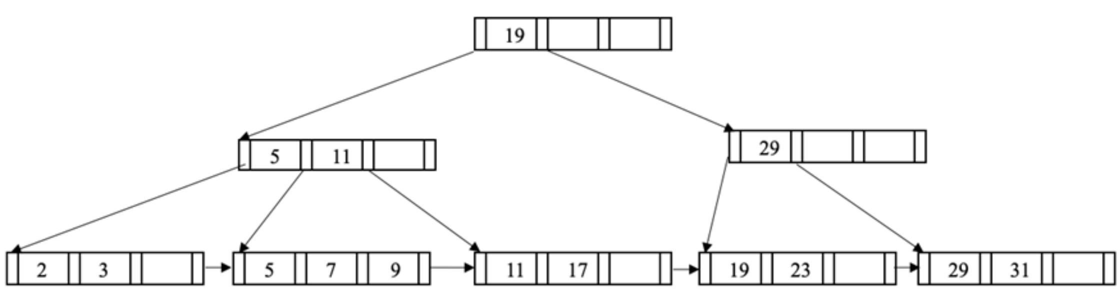
b.1. Insert 9
b.2. Insert 10
b.3. Insert 8
b.4. Delete 23
b.5. Delete 19

Solution:

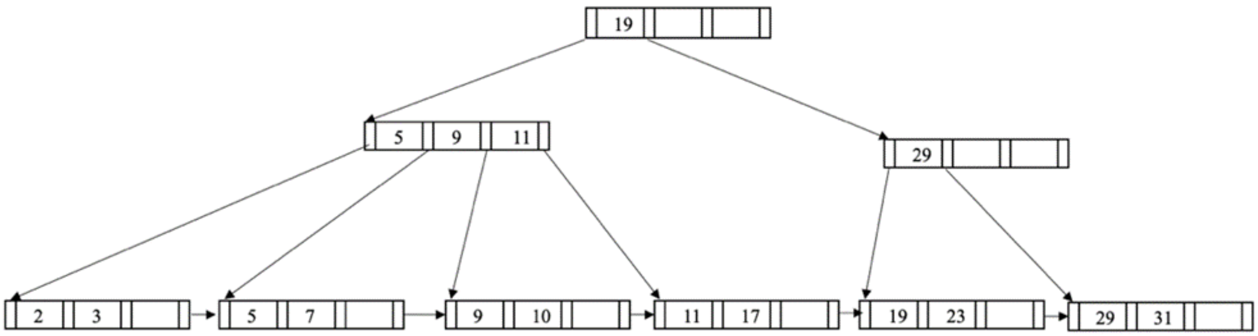
a. The tree is:



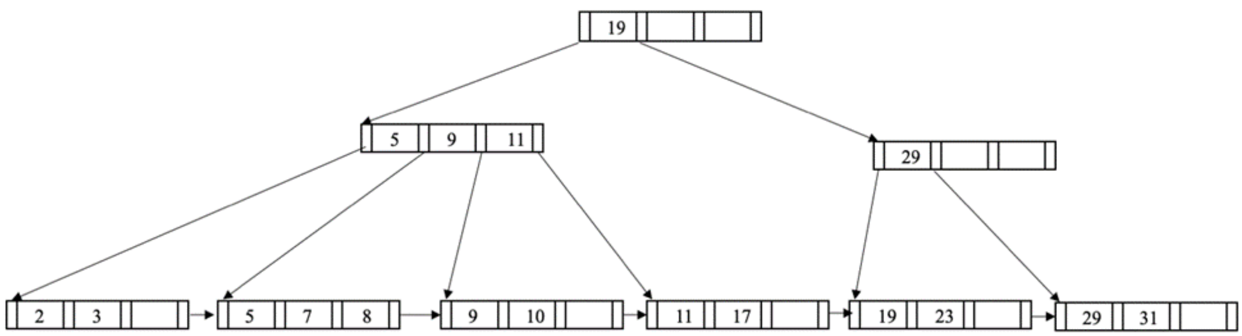

b.1. Insert 9, and the tree becomes:



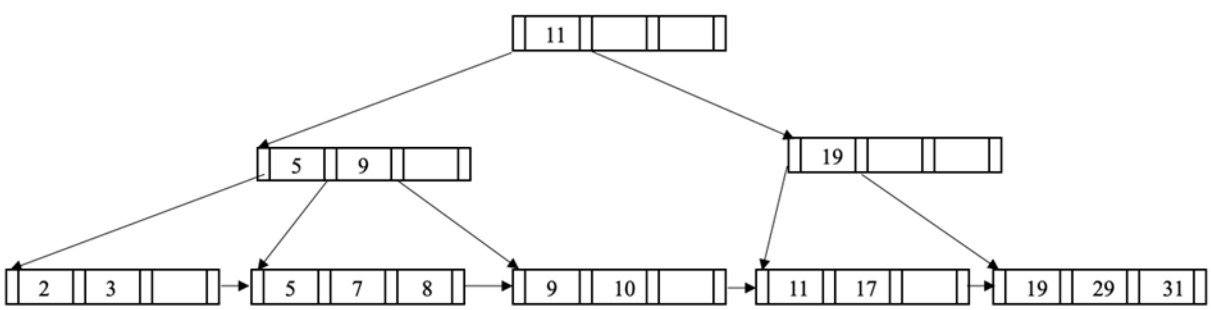
b.2. Insert 10, and the tree becomes:



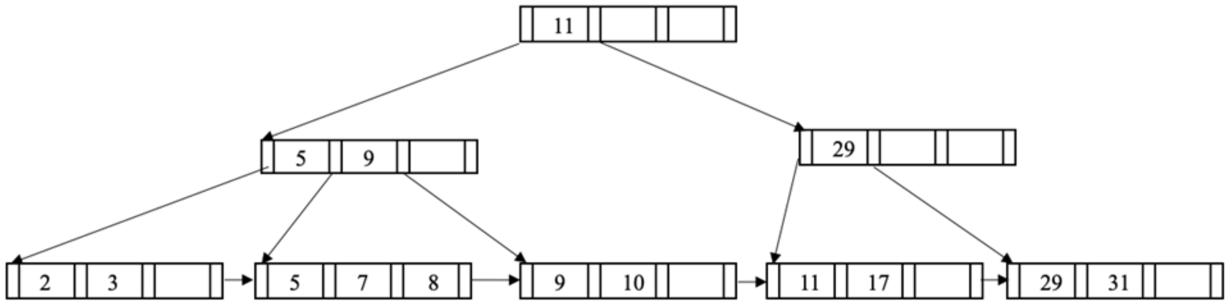

b.3. Insert 8, and the tree becomes:




b.4. Delete 23, and the tree becomes:




b.5. Delete 19, the tree becomes:

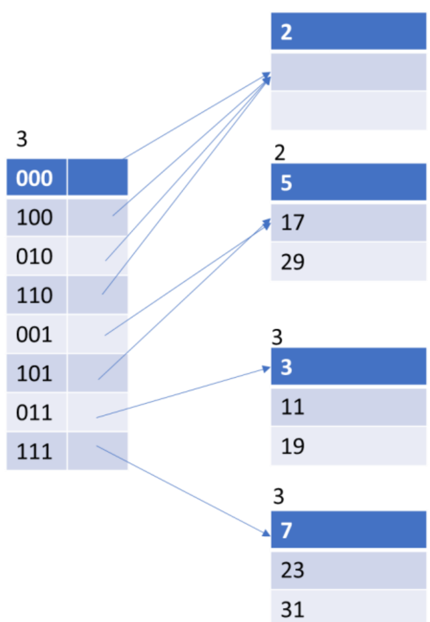



Problem 2. This problem has two parts:

a. Suppose that we are using extendable hashing on a file that contains records with the following search key values:

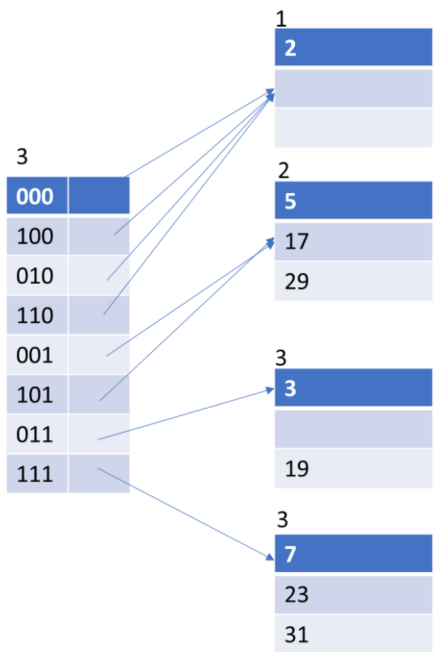
$$2, 3, 5, 7, 11, 17, 19, 23, 29, 31.$$

Show the extendable hash structure for this file if the hash function is $h(x) = x\%8$, and each bucket can hold three records.

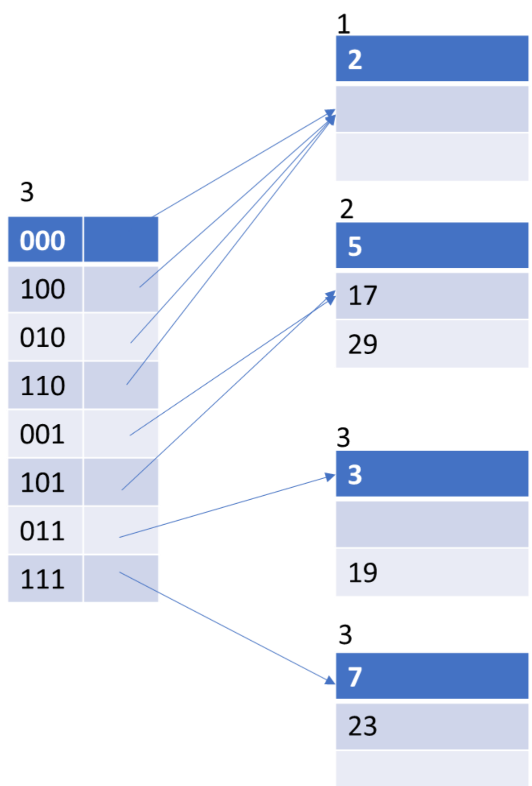



b. After Completing Part a, show the extendable hash structures after the following operations:

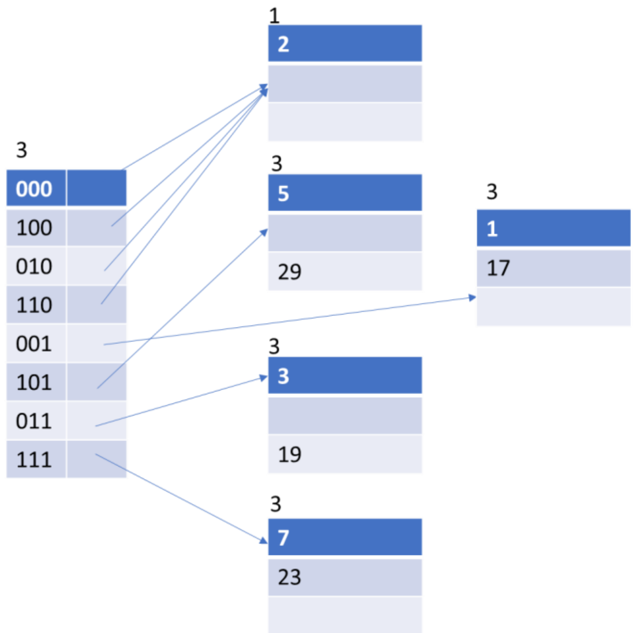
b.1. Delete 11


1
2
3
000
100
010
110
001
101
011
111
2
5
17
29
3
3
19
3
7
23
31

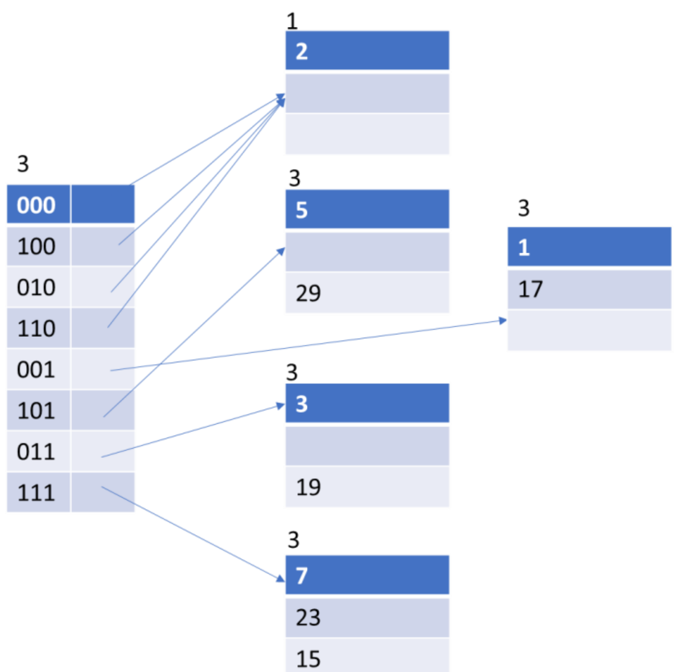
b.2. Delete 31


1
2
3
000
100
010
110
001
101
011
111
2
5
17
29
3
3
19
3
7
23

b.3. Insert 1




b.4. Insert 15




Problem 3. Consider the following SQL query for our University Database:

*select T.dept_name*
*from department as T, department as S*

*where T.budget > S.budget and S.building = "MAG"*

Write an efficient relational-algebra expression that is equivalent to this query. Justify your choice.

$$\Pi_{T.dept_name}(\rho_T(department) \bowtie\}_{T.budget>S.budget}$$

$$\rho_S \left(\sigma_{department.building=\mathrm{MAG}}(department)\right)$$

1. First, select departments located in "MAG" building, which shall reduce the departments significantly.
2. Do a conditional join to find departments with budgets greater than any department located in "MAG" building"
3. Extract the department names after step 2.

   Note: For execution, step2 and 3 can be combined.

Problem 4. Let the relations $r_1(A,B,C)$ and $r_2(C,D,E)$ have the following properties: $r_1$ has 20,000 tuples, $r_2$ has 45,000 tuples, 25 tuples of $r_1$ fit on one block, and 30 tuples of $r_2$ fit on one block. Estimate the number of block accesses required, using each of the following join strategies for $r_1 \rhd\lhd r_2$:

a. Nested-loop join
b. Block nested-loop join
c. Merge join
d. Hash join

$n_{r_1} = 20{,}000, n_{r_2} = 45{,}000$
$b_{r_1} = 20{,}000/25 = 800,\ \ b_{r_2} = 45{,}000/30 = 1{,}500.$

a. If $r_1$ is used for the outer loop, then the number of block transfers is 20,000 * 1500 + 800 = 30,000,800.

   If $r_2$ is used for the outer loop, then the number of block access would be 45,000 *800 + 1500 = 36,001,500.

   In either of the two cases, the number of disk seeks is $n_{r_1} + n_{r_2} = 800 + 1{,}500 = 2{,}300.$

b. If $r_1$ is used for the outer loop, then the number of block transfers is 800 * 1500 + 800 = 1,200,800. The number of disk seeks is $2 \times\ b_{r_1}\ =\ 1{,}600.$

If $r_2$ is used for the outer relation, then the number of block transfers is 1500 *800 + 1500 = 1,201,500. The number of disk seeks is $2 \times \mathrm{b}_{\mathrm{r}_2} = 3{,}000$.

c. Assume both relations are sorted. In the worst case, we need $b_{r_1} + b_{r_2}$ = 800 + 1500 = 2,300 block transfers. The number of disk seeks is also $b_{r_1} + b_{r_2}$ = 800 + 1500 = 2,300.

If the relations are not sorted, we need to add the cost for sorting the relations.

d. For hash join, the total number of block transfers is $3\times(b_{r_1} + b_{r_2})$ = 6,900 , and the total number of disk seeks is $2 \times (b_{r_1} + b_{r_2})$= 4,600. Here, we ignore the extra cost pertaining to the number of hash buckets which is determined by the specific hash functions used.

Problem 5. Show that the following equivalences hold.

a. $E_1 \bowtie_\theta (E_2 - E_3) = (E_1 \bowtie_\theta E_2 - E_1 \bowtie_\theta E_3)$

b. $\sigma_{\theta_1 \wedge \theta_2}(E_1 \bowtie_{\theta_3} E_2) = \sigma_{\theta_1}(E_1 \bowtie_{\theta_3} (\sigma_{\theta_2}(E_2)))$, where $\theta_2$ involves only attributes from $E_2$.

a. $t \in E_1 \bowtie_\theta (E_2 - E_3)$, if and only if there are $u \in E_1, v \in E_2 - E_3$, such that $u$ and $v$ satisfy $\theta$ and $t$ is the join of $u$ and $v$. Note that $v \in E_2 - E_3$ implies $v \in E_2$ but $v \notin E_3$. The latter condition is equivalent to $t \in E_1 \bowtie_\theta E_2, but\ t \notin E_1 \bowtie_\theta E_3$.

b. $\sigma_{\theta_1 \wedge \theta_2}(E_1 \bowtie_{\theta_3} E_2) = \sigma_{\theta_1}\left(\sigma_{\theta_2}(E_1 \bowtie_{\theta_3} E_2)\right) = \sigma_{\theta_1}(E_1 \bowtie_{\theta_3 \wedge \theta_2} E_2)$. Since $\theta_2$ involves only attributes from $E_2$, the latter is same as $\sigma_{\theta_1}(E_1 \bowtie_{\theta_3} (\sigma_{\theta_2}(E_2)))$.