Course 13

Problems

Assessment

- 20 questions, each with 4 answers. Only one answer is correct. (20%)
- 3 problems (30%)
- Laboratory activity/test (50%)

Problem 1

Consider the following instance of the Students relation:

sid	name	email	age	grade
2833	Jones	jones@scs.ubbcluj.ro	19	9
2877	Smith	smith@scs.ubbcluj.ro	20	8
2976	Jones	jones@math.ubbcluj.ro	21	10
2765	Mary	mary@math.ubbcluj.ro	22	7.7
3000	Dave	dave@cs.ubbcluj.ro	18	5.5
3010	Smith	smith2@scs.ubbcluj.ro	20	7
3020	Sam	sam@scs.ubbcluj.ro	19	9.5

1. Give an example of an attribute (or set of attributes) that you can deduce is not a candidate key, based on this instance being legal.
2. Is there any example of an attribute (or set of attributes) that you can deduce is a candidate key, based on this instance being legal?

Problem 1 - Solution

1. Examples of non-candidate key include the following: *name*, *age*. (*grade* can not be declared a non-candidate key from this evidence alone, even though the common sense tells us that clearly more than one student could have the same grade).

2. We cannot determine a key of a relation given only one instance of the relation. The fact that the instance is "legal" is immaterial. The instance shown is just one possible "snapshot" of the relation

Problem 2

Consider the following movie database schema (types / domains are omitted)

- *Actors* (ID, Name, YoB),where *ID* is a unique reference number for an actor, *Name* and *YoB* the name and year of birth of the actor;

- *Movies*(ID, Title, Year), where *ID* is a unique reference number for a movie, *Title* and *Year* the title and the year of production of the movie;

- *Casting*(MovieID, ActorID, Charac), where *MovieID* and *ActorID* are the reference numbers of movie and an actor. The actor plays the character, *Charac,* in the movie.

Express the following queries in SQL:

Problem 2 - Solution

1. *Find the titles of the movies produced after 1950 and before 2000;*

```
SELECT Title FROM Movies
WHERE Year > 1950 AND Year < 2000
```

2. *Find the titles of the movies produced after 2000 or before 1950;*

```
SELECT Title FROM Movies
WHERE Year < 1950 OR Year > 2000
```

Problem 2 - Solution

3. *Find the names of the actors playing "Fletcher Christian" in a production of "Mutiny on the Bounty" as well as the production year;*

```
SELECT Actors.Name, Movies.Year
FROM Actors
     INNER JOIN Casting
          ON Casting.ActorID = Actors.ID
     INNER JOIN Movies
          ON Casting.MovieID = Movies.ID
WHERE Casting.Charac = 'Fletcher Christian'
  AND Movies.Title = 'Mutiny on the Bounty'
```

Problem 2 - Solution

4. *Find the names of the actors who played "Superman" and "Clark Kent" in the same production of a movie;*

```
SELECT DISTINCT Actors.Name
FROM Actors
     INNER JOIN Casting as C1
          ON C1.ActorID = Actors.ID
     INNER JOIN Casting as C2
          ON C2.ActorID = Actors.ID
WHERE    C1.Charac = 'Superman'
     AND C2.Charac = 'Clark Kent'
     AND C1.MovieID = C2.MovieID
```

Problem 2 - Solution

5. *Find the names of the actors who played two different characters in the same production of a movie;*

```
SELECT DISTINCT Actors.Name
FROM Actors
     INNER JOIN Casting as C1
          ON C1.ActorID = Actors.ID
     INNER JOIN Casting as C2
          ON C2.ActorID = Actors.ID
WHERE    C1.Charac <> C2.Charac
     AND C1.MovieID = C2.MovieID
```

Problem 2 - Solution

6. *Find the pairs of names of the different actors who played the same character in different productions of the same movie.*

```
SELECT A1.Name, A2.Name
FROM Actors A1, Actors A2
INNER JOIN Casting C1 ON C1.ActorID = A1.ID
INNER JOIN Movies M1 ON C1.MovieID = M1.ID
INNER JOIN Casting C2 ON C2.ActorID = A2.ID
INNER JOIN Movies M2 ON C2.MovieID = M2.ID
WHERE M1.Title = M2.Title AND
      M1.Year <> M2.Year AND
      C1.Charc = C2.Charac AND A1.ID<>A2.ID
```

Problem 2 - Solution

7. Find the name of the actors who played in a movie with Victoria Abril and were born after she played her first movie

```
SELECT Actors.Name FROM Actors WHERE
Actors.ID IN
(SELECT A1.ID FROM Actors A1, Actors A2
INNER JOIN Casting C1 ON C1.ActorID = A1.ID
INNER JOIN Casting C2 ON C2.ActorID = A2.ID
WHERE A2.Name='Victoria Abril' AND
      C2.MovieID=C1.MovieID AND A1.ID<>A2.ID)
AND Actors.YoB > ANY
(SELECT Min(M.Year)FROM Movies M
INNER JOIN Casting C ON C.MovieID = M.ID
INNER JOIN Actors A ON C.ActorID = A.ID
WHERE A.Name = 'Victoria Abril')
```

Problem 3

Consider the following schema:

Suppliers (sid, sname, address)

Products (pid, pname, color)

Catalog (sid, pid, cost)

The key fields are underlined. The Catalog relation lists the prices charged for products by suppliers.

State what the following 6 queries compute:

Problem 3 - Solution

1. $\pi_{sname}(\pi_{sid}((\sigma_{color='red'}\ Products) \otimes (\sigma_{cost<100}\ Catalog)) \otimes\ Suppliers)$

Find the Supplier names of the suppliers who supply a red product that cost less than 100.

2. $\pi_{sname}(\pi_{sid}((\sigma_{color='red'}\ Products) \otimes (\sigma_{cost<100}\ Catalog) \otimes\ Suppliers))$

The RA statement does not return anything because of the sequence of projection operators. Once the *sid* is projected, it is the only field in the set.

Problem 3 - Solution

3. $\pi_{sname}((\sigma_{color='red'} Products) \otimes (\sigma_{cost<100} Catalog) \otimes Suppliers) \cap \pi_{sname}((\sigma_{color='green'} Products) \otimes (\sigma_{cost<100} Catalog) \otimes Suppliers)$

Find the Supplier names of the suppliers who supply at least a red product that cost less than 100 and at least a green product that cost less than 100 .

4. $\pi_{sname}((\sigma_{color='red'} Products) \otimes (\sigma_{cost<100} Catalog) \otimes Suppliers) \cup \pi_{sname}((\sigma_{color='green'} Products) \otimes (\sigma_{cost<100} Catalog) \otimes Suppliers)$

Find the Supplier names of the suppliers who supply red or green products that cost less than 100.

Problem 3 - Solution

5. $(\pi_{sid,\ pid}\ Catalog)\ /\ \pi_{pid}\ Products$

(R_1/R_2 contains all x such that for <u>*every*</u> y in R_2, there is an xy in R_1)

Find the sids of of suppliers who supply every product.

6. $\rho(R_1, \text{Catalog}), \rho(R_2, \text{Catalog})$

$\pi_{R1.pid}((\sigma_{R1.pid = R2.pid \wedge R1.sid <> R2.sid}(R_1 \times R_2))$

Find *pids* of products that are supplied by at least two different suppliers

Problem 4

For the registration of books in our library, we need:

bookno, ISBN, section, serial no, sample no, title, author.

Each book (*sample*) has a unique number (*bookno*) being used for the registration of loans.This number is attached to the book as a barcode. This number is meaningless.

The notion of ISBN is supposed to be well known.

Each book belongs to a section. Within a section, different books have different serial numbers. If we have more than one sample of the same book, then these books are distinguished by sample numbers.

Which of the following FDs hold? Give a short explanation of your answers.

Problem 4 - Solution

1. $bookno \rightarrow \text{ISBN}, section, serial\ no, sample\ no$

Yes: each book (*sample*) has a unique number (*bookno*); *bookno* is a key

2. $\text{ISBN} \rightarrow bookno$

No: you may have several samples of a book

3. $\text{ISBN} \rightarrow section, serial\ no$

Yes: this constraint forces the librarian to be consistent

Problem 4 - Solution

4. ISBN $\rightarrow$ *sample no*

No: you may have several samples of a book

5. ISBN $\rightarrow$ *title, author*

Yes: From the definition of ISBN, each edition of a book has a specific ISBN. So, there are not different *titles* (and *authors*) whith the same ISMBN.

6. *sample no* $\rightarrow$ ISBN, *bookno*

No: the bare sample number does not provide information about the identity of the book.

Problem 5

Suppose that we have the following three tuples in a legal instance of a relation schema S with three attributes A, B, C listed in order:

(1,2,3)

(4,2,3)

(5,3,3)

1. Which of the following dependencies can you infer does *not* hold over schema S?

a) $A \rightarrow B$ b) $BC \rightarrow A$ c) $B \rightarrow C$

2. Can you identify any dependencies that hold over S?

Problem 5 - Solution

1. *Which of the following dependencies can you infer does not hold over schema S?*

   a) $A \rightarrow B$ -

   b) $BC \rightarrow A$ does not hold (1,2,3) & (4,2,3): different values for A with the same values for BC

   c) $B \rightarrow C$ -

2. *Can you identify any dependencies that hold over S?*

No. To say that a FD holds on a relation is to make a statement about *all* allowable instances of that relation.

Problem 6

Consider schema R(A;B;C;D;E) with FDs

F = {AB $\rightarrow$ CDE; AC $\rightarrow$ BDE; B $\rightarrow$ C; C $\rightarrow$ B; C $\rightarrow$ D; B $\rightarrow$ E}.

1. Find all keys of R.
2. Find a minimal cover of F.
3. Is R in BCNF? Explain.
4. Find a lossless-join BCNF decomposition of R.
5. Is your solution in part 4 a dependency-preserving decomposition? Explain.
6. Is R in 3NF? Explain.
7. Use the synthesis approach to compute a 3NF decomposition of R

Problem 6 - Solution

1. *Find all keys of R.*

$F = \{AB \rightarrow CDE; AC \rightarrow BDE; B \rightarrow C; C \rightarrow B; C \rightarrow D; B \rightarrow E\}$

$A^+ = A$

$B^+ = BCED$

- it is not possible to deduce A from the other attributes $\Rightarrow$ A belongs to key

$AB^+ = ABCDE$ - a candidate key

$AC^+ = ACBDE$ - a candidate key

$AD^+ = AD$,

$AE^+ = AE$

$ADE^+ = ADE$

Problem 6 - Solution

2. *Find a minimal cover of F.*

- A minimal cover for a set F of FDs is a set G of FDs such that
  1. Every FD in G is of the form $\alpha \to A$
  2. For each FD $\alpha \to A$ in G, α has no redundant attributes
  3. There are no redundant FDs in G
  4. G and F are equivalent
- Algorithm to compute minimal cover of F:
  1. Use decomposition rule to obtain FDs with 1 attr. on R.H.S.
  2. Remove redundant attributes from the L.H.S. of each FD
  3. Remove redundant FDs

Minimal cover of F:

$\{B \to C; C \to B; C \to D; B \to E\}$

Problem 6 - Solution

3. *Is R in BCNF? Explain.*

- Relation R with FDs F is in BCNF if, for all $\alpha \rightarrow A$ in F^+
  - $A \in \alpha$ (*trivial* FD), or
  - α contains a key for R (α is a superkey).

$F = \{AB \rightarrow CDE; AC \rightarrow BDE; B \rightarrow C; C \rightarrow B; C \rightarrow D; B \rightarrow E\}$

candidate keys: AB and AC

R is not in BCNF because there is at least one FD (e.g. $B \rightarrow C$) which is not trivial and the left side is not a superkey.

Problem 6 - Solution

4. *Find a lossless-join BCNF decomposition of R.*

- Consider relation R with FDs F. If $\alpha \rightarrow A$ violates BCNF, decompose R into R - A and αA.

F = {AB $\rightarrow$ CDE; AC $\rightarrow$ BDE; B $\rightarrow$ C; C $\rightarrow$ B; C $\rightarrow$ D; B $\rightarrow$ E}

B $\rightarrow$ C violates BCNF

Decompose R in {**AB**DE}, {**B**C}

B $\rightarrow$ E violates BCNF

Decompose R in {**AB**D}, {**B**C}, {**B**E}

Problem 6 - Solution

5. *Is your solution in part 4 a dependency-preserving decomposition? Explain.*

$F = \{AB \rightarrow CDE; AC \rightarrow BDE; B \rightarrow C; C \rightarrow B; C \rightarrow D; B \rightarrow E\}$

Our decomposition: {**AB**D}, {**B**C}, {**B**E}

The resulting decomposition is BCNF, is lossless-join but is not dependency preserving (in order to preserve dependencies redundant relations should be added {CD} and {CB}.

Problem 6 - Solution

6. *Is R in 3NF? Explain.*

- Relation R with FDs F is in 3NF if, for all $\alpha \rightarrow A$ in F^+
  - $A \in \alpha$ (*trivial* FD), or
  - α is a superkey for R, or
  - A is part of some key for R (i.e. A is a prime attribute).

F = {AB $\rightarrow$ CDE; AC $\rightarrow$ BDE; B $\rightarrow$ C; C $\rightarrow$ B; C $\rightarrow$ D; B $\rightarrow$ E}

candidate keys: AB and AC

R is not in 3NF because there is at least one FD (e.g. C $\rightarrow$ D) which is not trivial, the left side is not a superkey and D $\not\subset$ AB or D $\not\subset$ AC.

Problem 6 - Solution

7.Use the synthesis approach to compute a 3NF decomposition of R

Input: Schema R with FDs *F* which is a minimal cover

Output: A dependency-preserving, lossless-join 3NF dec. of R

Initialize D = $\varnothing$

Apply *union rule* to combine FDs in F with same attribute in he left side into a single FD

For each FD $\alpha \rightarrow \beta$ in F do

Insert the relation schema $\alpha\beta$ into D

Insert δ into D, where δ is some key of R

Remove redundant relation schema from D as follows:

delete R_i from D if $R_i \subseteq R_j$, where $R_j \in D$

return D

Problem 6 - Solution

$\{B \rightarrow C; C \rightarrow B; C \rightarrow D; B \rightarrow E\}$ - minimal cover
AB, AC candidate keys

use union rule to combine FDs with the same LHS $\Rightarrow$
$\{B \rightarrow CE, C \rightarrow BD\}$

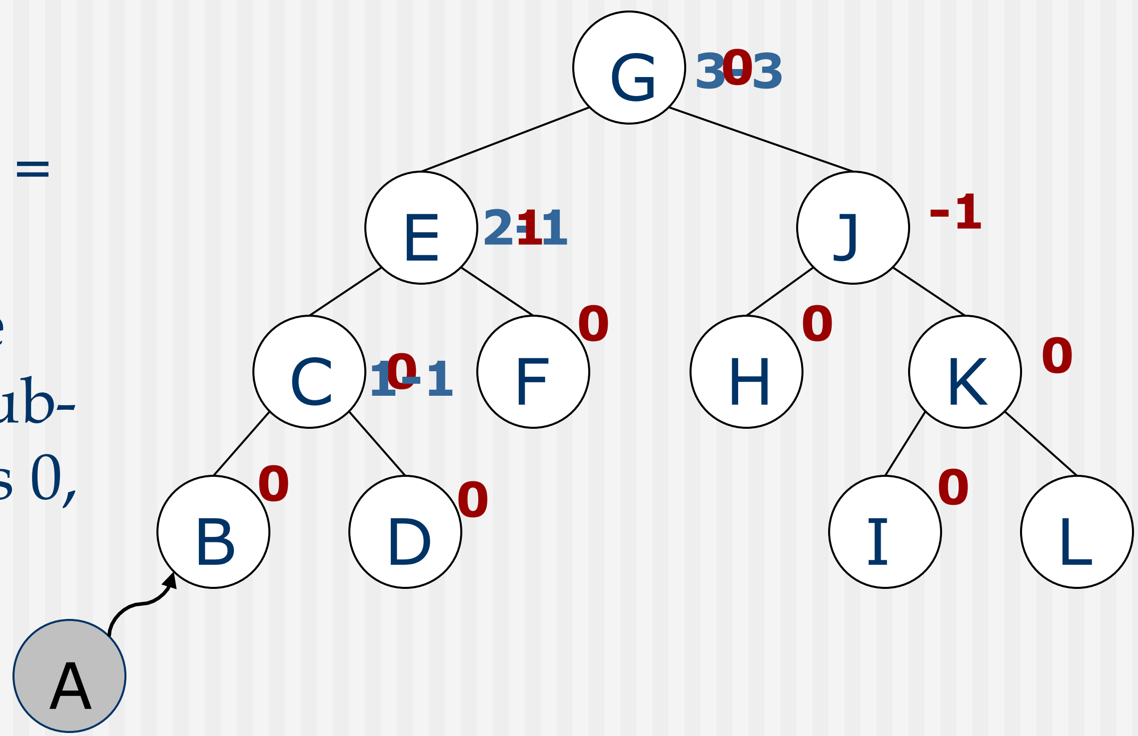
Decomposition of R:

{**B**;C;E}
{**C**;B;D} > generated from FDs
{**A;B**}
{**A;C**} > generated from keys

Problem 7

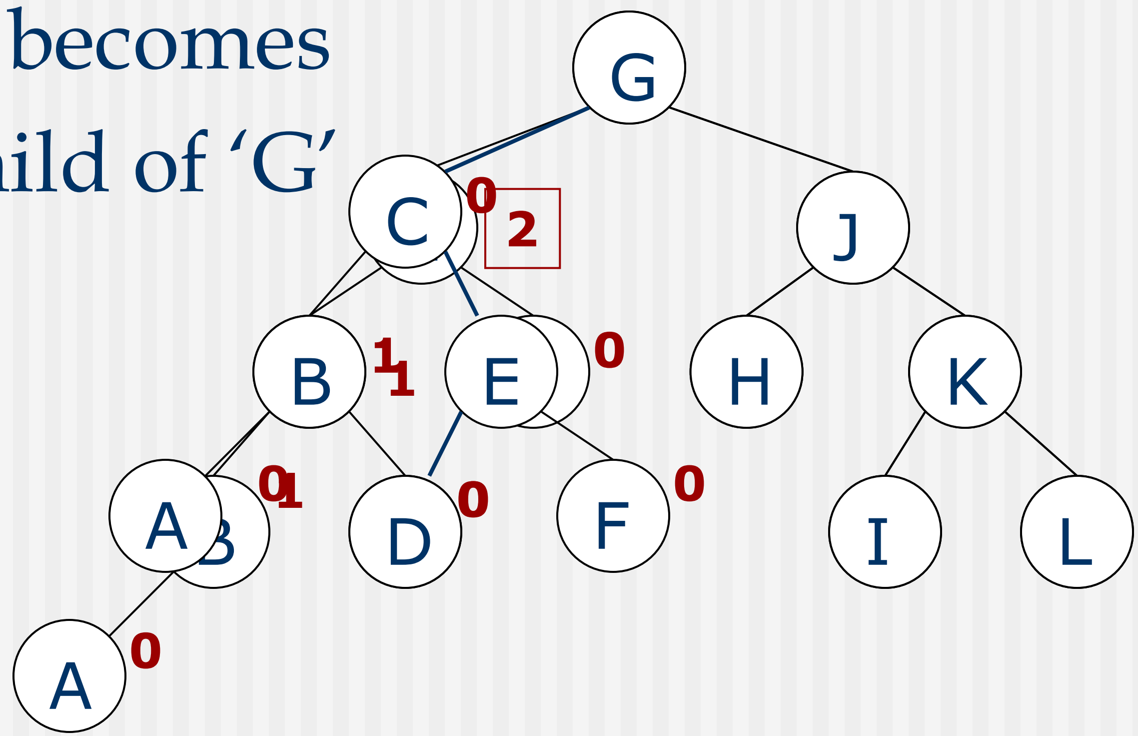
Consider the following balanced binary tree. Describe the operations needed to keep the tree balanced after insertion a new record with key value '*A*'

Balanced tree = for each node the difference between its sub-trees *heights* is 0, 1 or –1



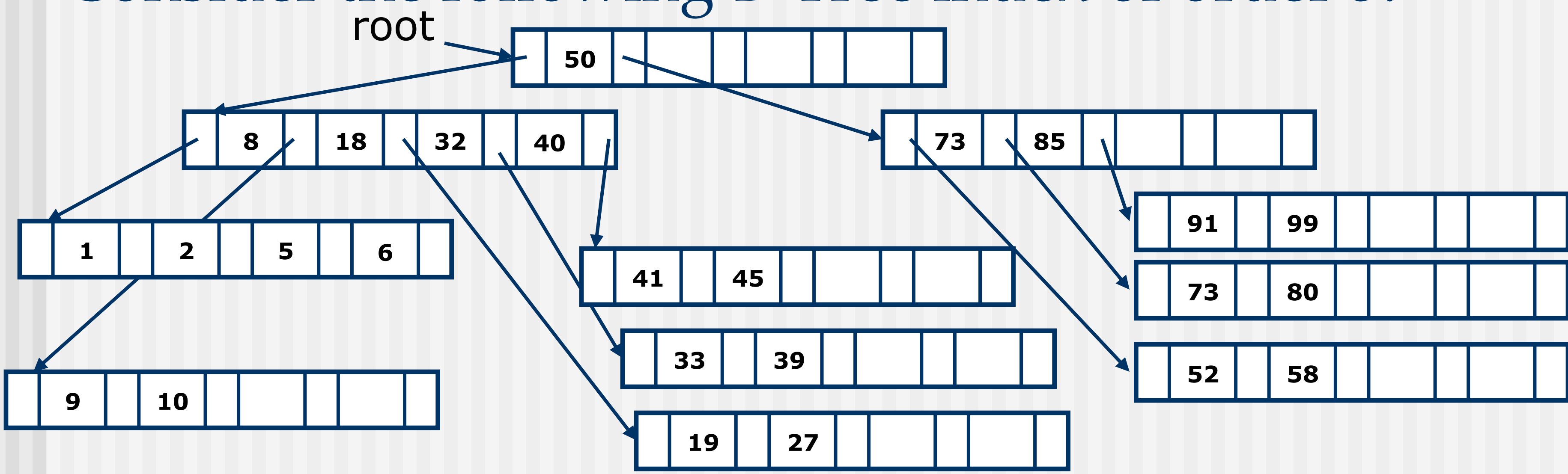
Problem 7 - Answer

- node ‘E’ becomes the right child of ‘C’
- node ‘D’ becomes the left child of ‘E’
- node ‘C’ becomes the left child of ‘G’



Problem 8

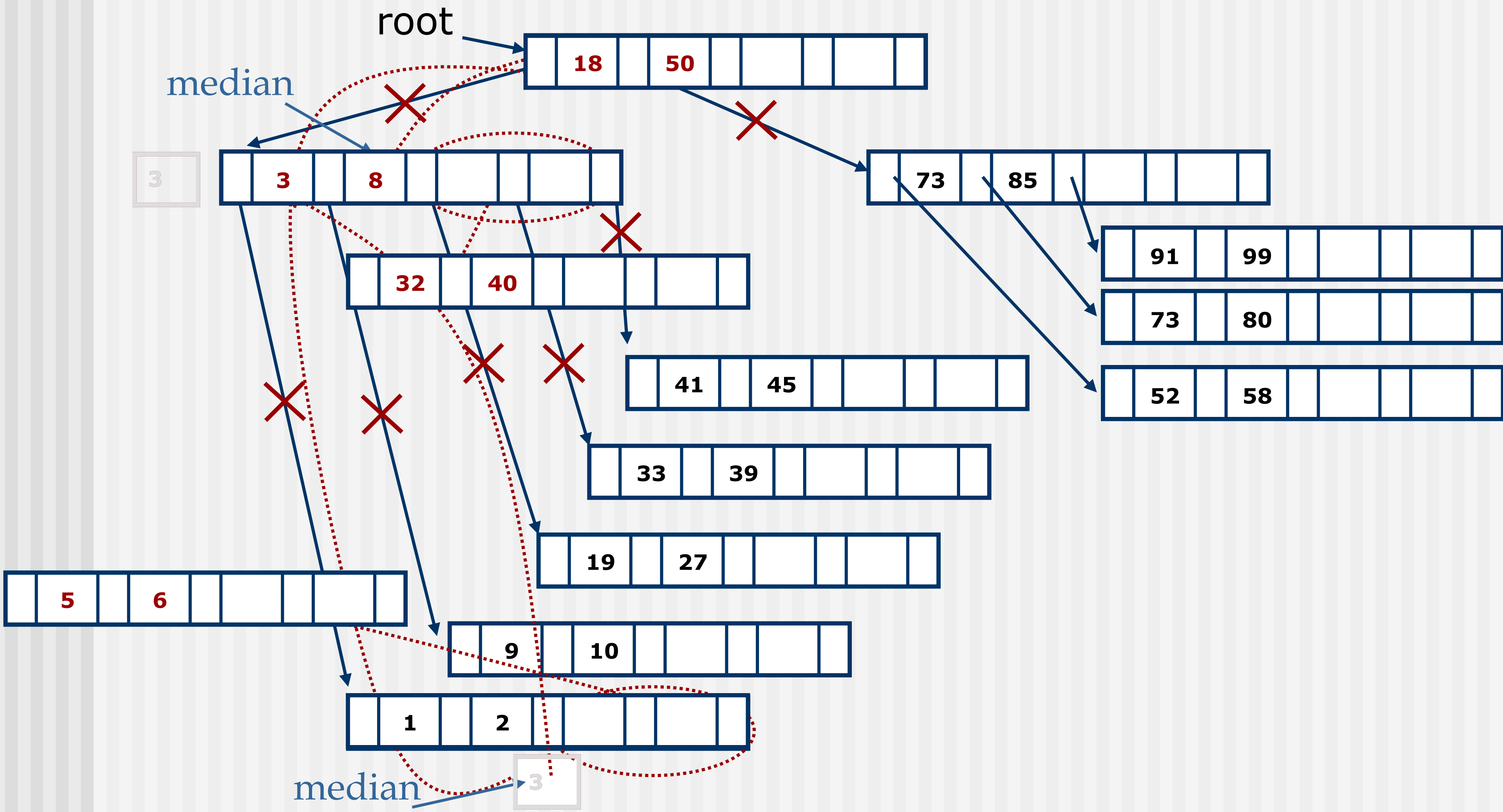
Consider the following B-Tree index of order 5:



1. Show the B-Tree that would result from inserting a data entry with key '3' into this tree.
2. Show the B-Tree that would result from deleting the data entry with key '8' from the original tree using redistribution.
3. Show the B-Tree that would result from deleting the data entry with key '8' from the original tree with merging.

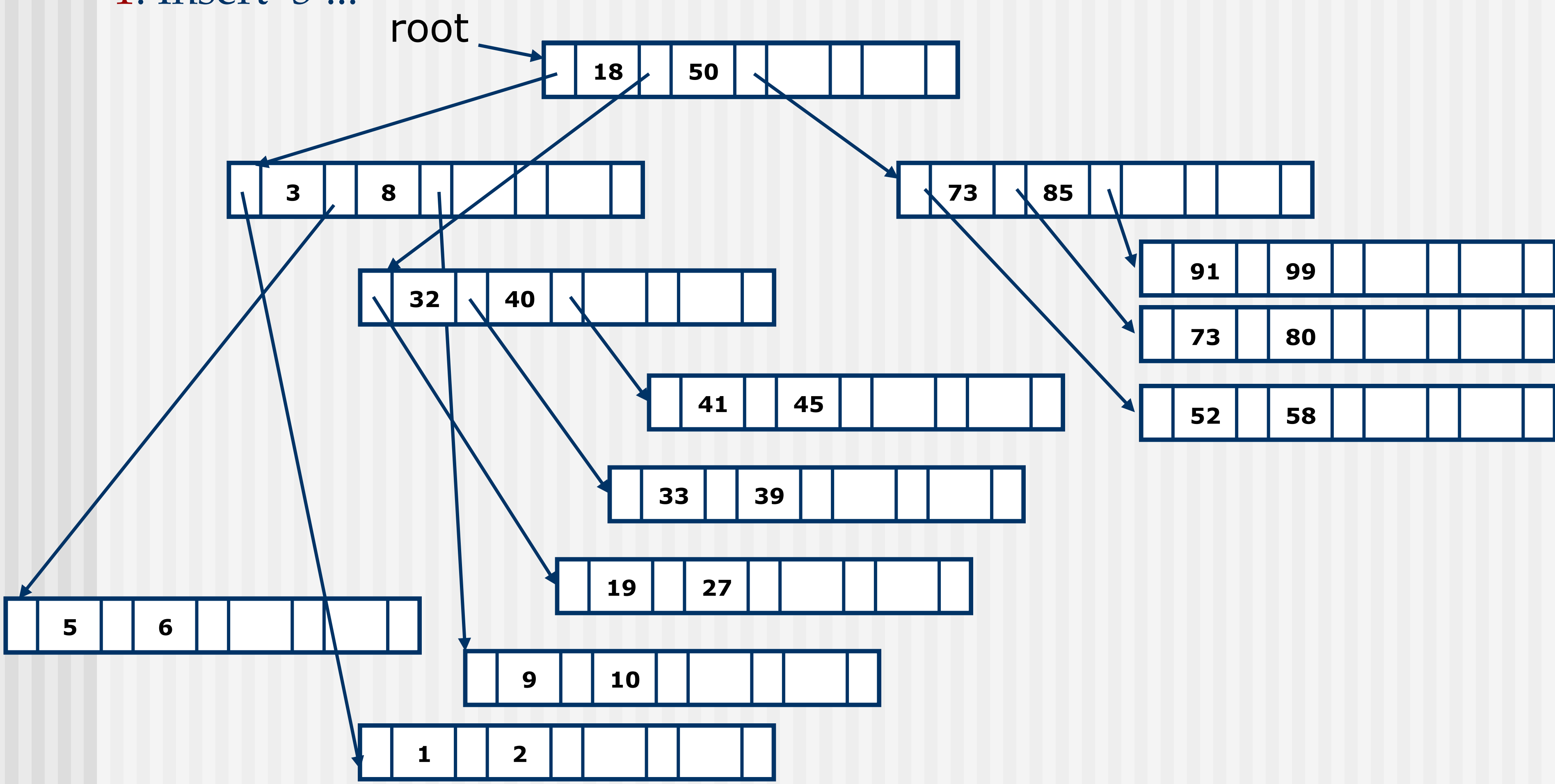
Problem 8 - Answer

1. Insert '3'...



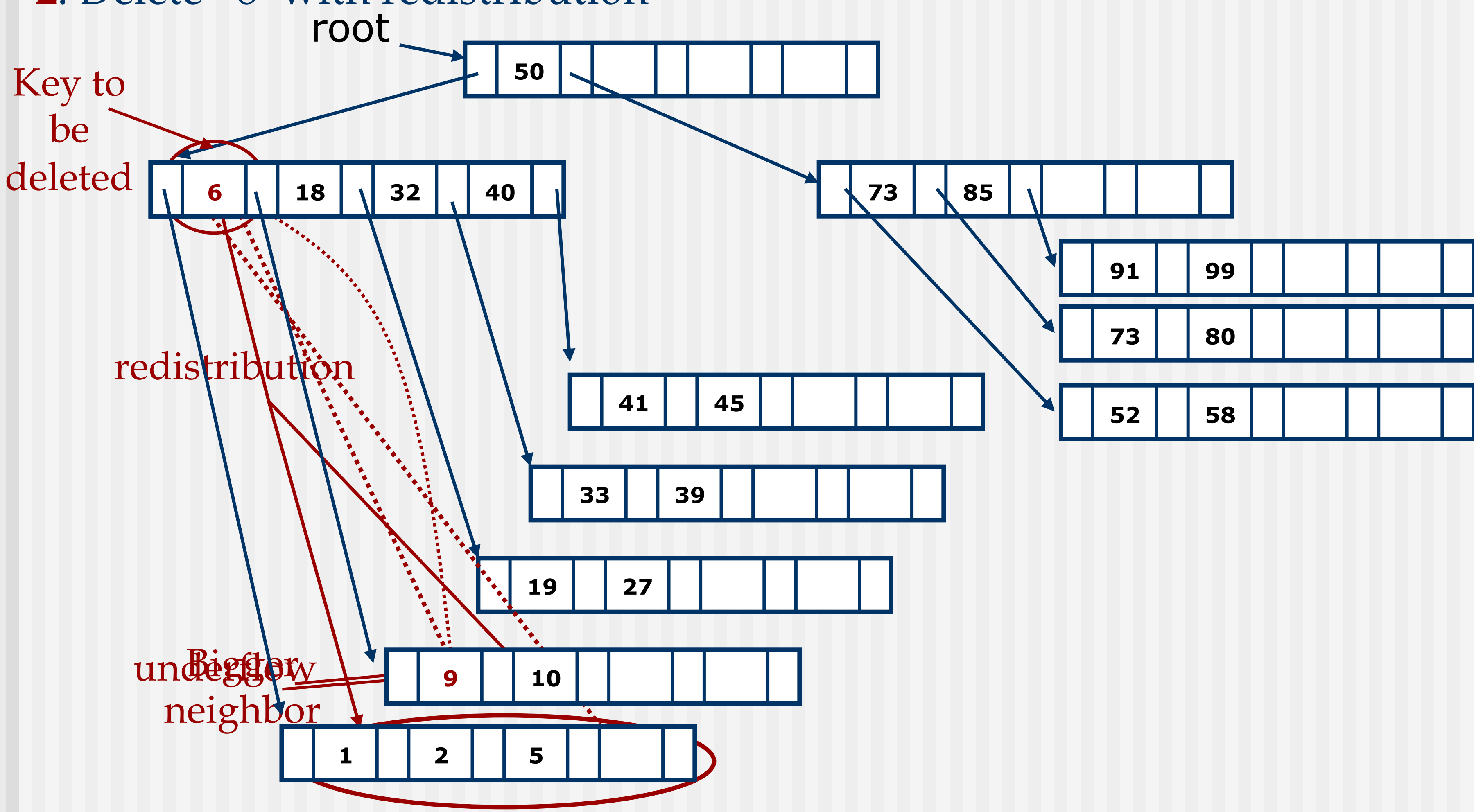
Problem 8 – Answer

1. Insert '3'…



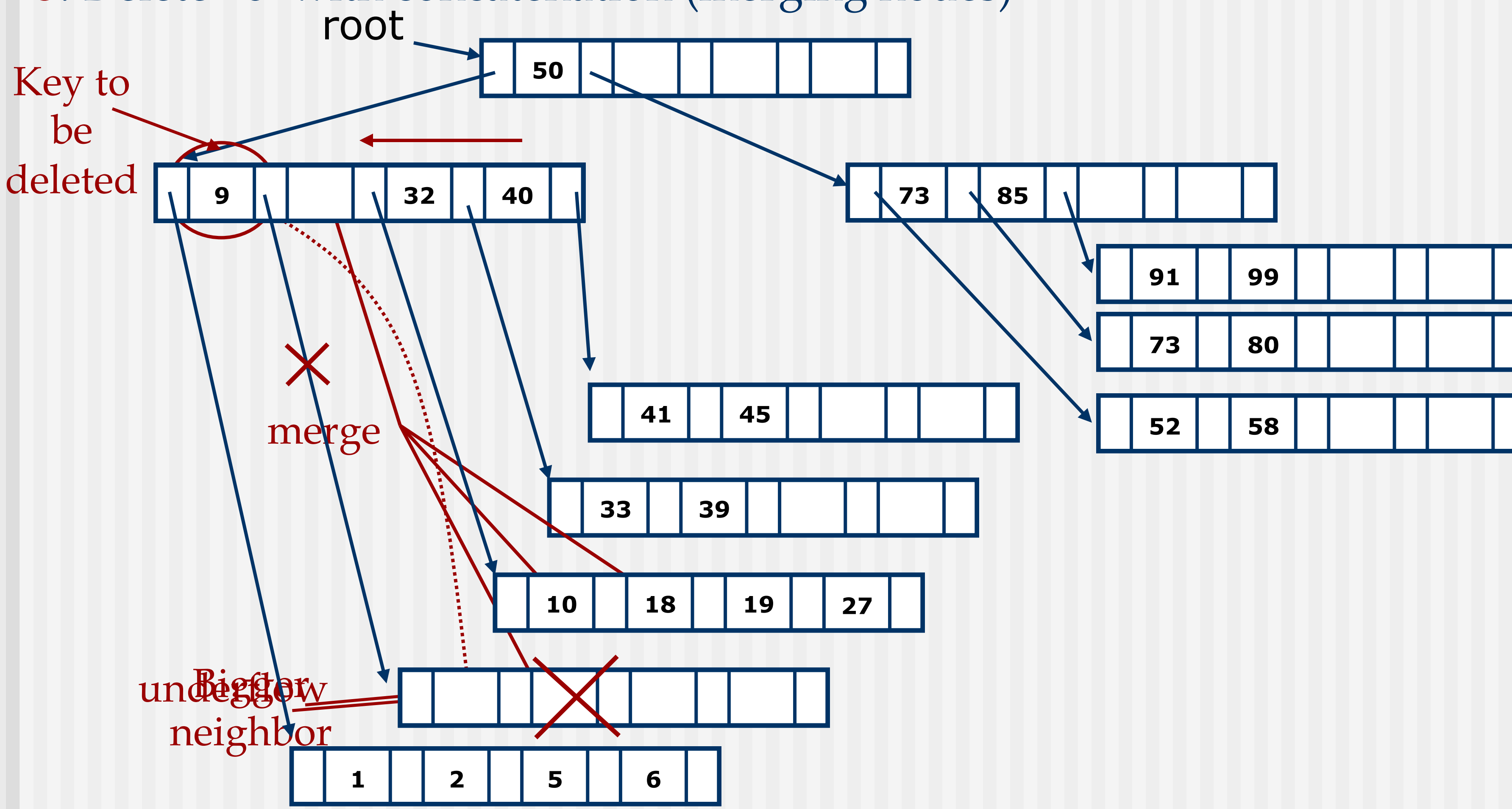
Problem 8 – Answer (cont)

2. Delete ‘8’ with redistribution



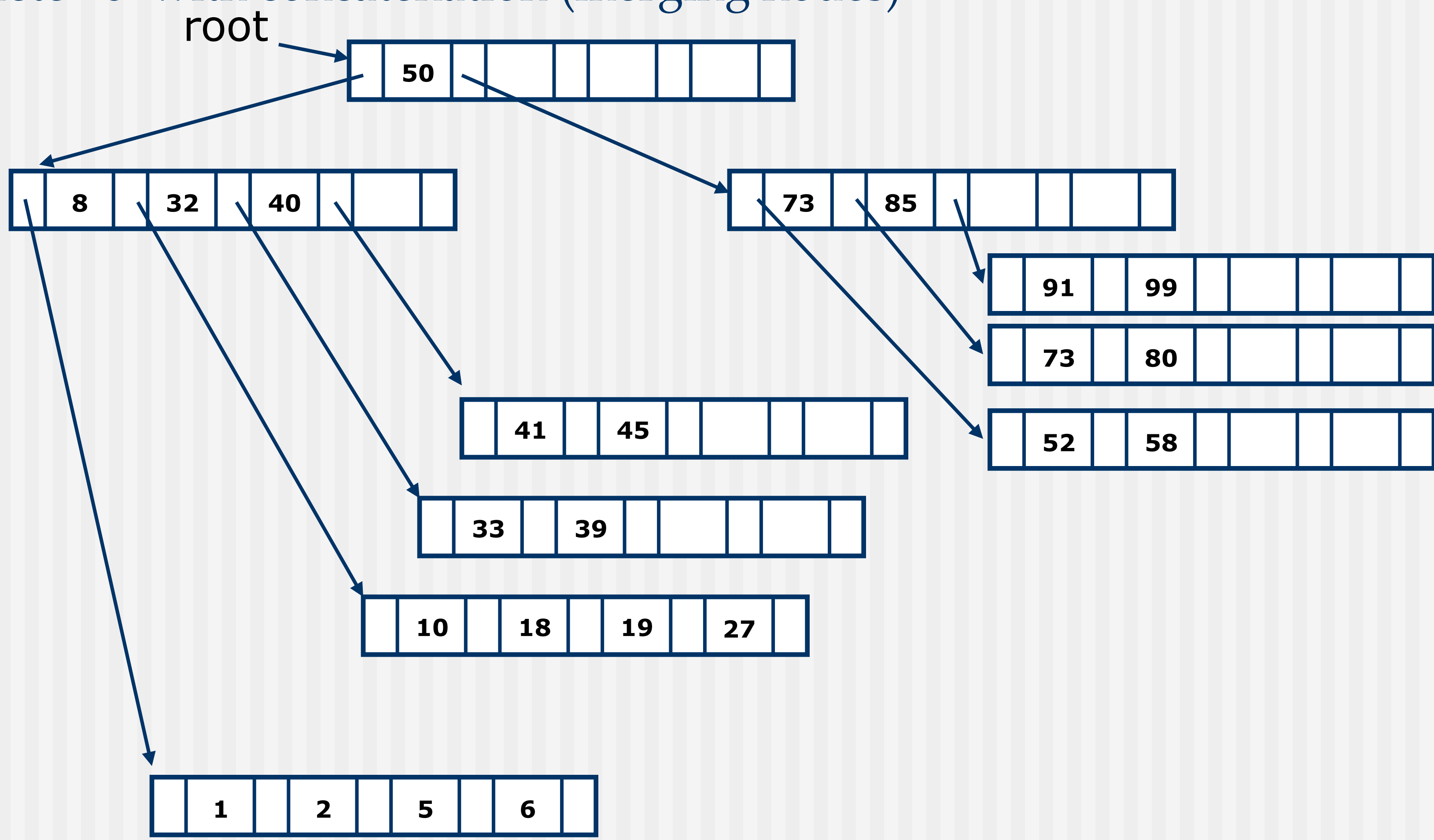
Problem 8 – Answer (cont)

3. Delete ‘8’ with concatenation (merging nodes)



Problem 8 – Answer (cont)

3. Delete ‘8’ with concatenation (merging nodes)



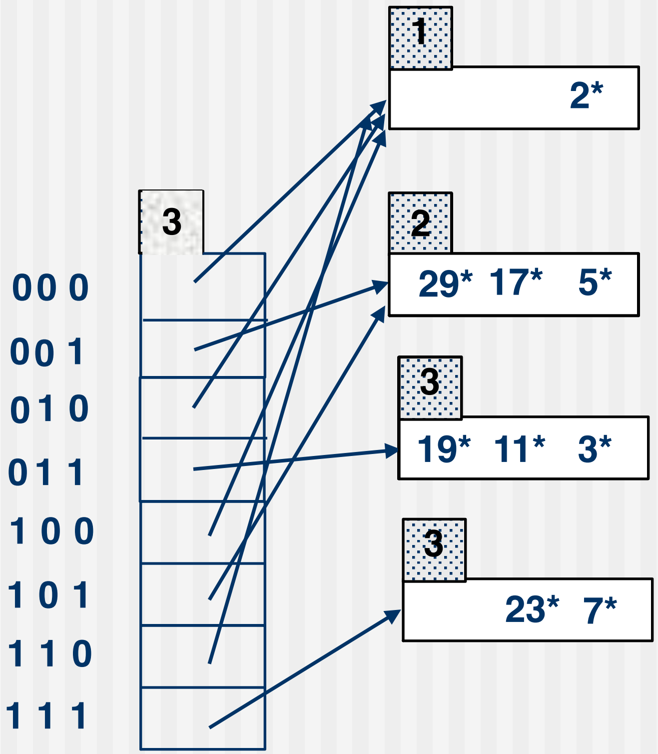
Problem 9

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.

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

Problem 9 - Answer



x	x mod 8	bin
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011
17	1	001
19	3	011
23	7	111
29	5	101

Problem 10

Consider a disk with a sector size of *512* bytes, *2.000* tracks per surface, *50* sectors per track, *5* double-sided platters, avg. seek time of *10* msec.

1. What is the capacity of a track (in bytes)? What is the capacity of each surface? What is the capacity of the disk?
2. How many cylinders does the disk have?
3. Is 256 bytes a valid block size? 2.048? 51.200?
4. If the disk platters rotate at 5.400 rpm, what is the maximum rotational delay?
5. Assuming that one track of data can be transferred per revolution, what is the transfer rate?

Problem 10 - Answers

1. *What is the capacity of a track (in bytes)? What is the capacity of each surface? What is the capacity of the disk?*

bytes/track = bytes/sector * sector/track = 512*50 = 25*Kb*

bytes/surface = bytes/track * track/surface = 25 *Kb* * 2000 =
= 50000 *Kb*

bytes/disk = bytes/surface*surfaces = 50.000*Kb* * 2*5 =
= 500000 *Kb*

2. *How many cylinders does the disk have?*

no. of cylinders = no. of tracks on each platter $\rightarrow$ 2000

Problem 10 – Answers (cont)

3. *Is 256 bytes a valid block size? 2.048? 51.200?*

Block size is a multiple of sector size → only *2.048* and *51.200* are valid block sizes.

4. *If the disk platters rotate at 5.400 rpm, what is the maximum rotational delay?*

Maximum rotational delay = time required for a rotation

60 / 5400 = 0,011 sec.

5. *Assuming that one track of data can be transferred per revolution, what is the transfer rate?*

25 Kb / 0,011 = 2.250 bytes per sec