

Übungen

# Aufgabe 1

- Wir haben folgende Ausprägung der Relation Studenten:

SID	Name	Email	Age	Note
2833	Jones	<a href="mailto:jones@scs.ubbcluj.ro">jones@scs.ubbcluj.ro</a>	19	9
2877	Smith	<a href="mailto:smith@scs.ubbcluj.ro">smith@scs.ubbcluj.ro</a>	20	8
2976	Jones	<a href="mailto:jones@math.ubbcluj.ro">jones@math.ubbcluj.ro</a>	21	10
2765	Mary	<a href="mailto:mary@math.ubbcluj.ro">mary@math.ubbcluj.ro</a>	22	7.7
3000	Dave	<a href="mailto:dave@cs.ubbcluj.ro">dave@cs.ubbcluj.ro</a>	18	5.5
3010	Smith	<a href="mailto:smith2@scs.ubbcluj.ro">smith2@scs.ubbcluj.ro</a>	20	7
3020	Sam	<a href="mailto:sam@scs.ubbcluj.ro">sam@scs.ubbcluj.ro</a>	19	9.5

1. Finde ein Attribut oder Menge von Attributen, die kein Kandidatschlüssel sein können, bzgl. dieser Ausprägung
2. Können wir einen Kandidatschlüssel für die Relation Studenten finden, wenn wir wissen dass diese Ausprägung gültig ist?

# Aufgabe 1 - Lösung

1. Nicht-Kandidatschlüssel: Name, Age

Note kann bzgl. dieser Ausprägung nicht als Nicht-Kandidatschlüssel identifiziert werden. In dieser Ausprägung ist Note eindeutig, obwohl wir wissen, dass es normalerweise nicht eindeutig ist.

2. Nein, wir können den Kandidatschlüssel basierend auf einer Ausprägung nicht bestimmen.

## Aufgabe 2

Die Datenbank enthält folgende Relationen:

- Schauspieler(ID, Name, Geburtsjahr)
- Filme(ID, Titel, Jahr), wobei Jahr = wann der Film produziert wurde
- Casting(FilmId, SchauspielerID, Charakter) – ein Schauspieler spielt ein Charakter in einem Film

Die Primärschlüssel sind unterstrichen.

Schreibe folgenden Anfragen in SQL und/oder relationale Algebra

## Aufgabe 2

1. Finde die Titeln der Filme produziert zwischen 1950 und 2000 (SQL+rel.A.)

```
SELECT [DISTINCT] Titel FROM Filme  
WHERE Jahr > 1950 AND Jahr < 2000
```

$$\Pi_{\text{Titel}} (\sigma_{\text{Jahr} > 1950 \wedge \text{Jahr} < 2000} (\text{Filme}))$$

2. Finde die Titeln der Filme produziert nach 2000 oder bevor 1950(SQL+rel.A.)

```
SELECT [DISTINCT] Titel FROM Filme  
WHERE Jahr < 1950 OR Jahr > 2000
```

$$\Pi_{\text{Titel}} (\sigma_{\text{Jahr} < 1950 \vee \text{Jahr} > 2000} (\text{Filme}))$$

## Aufgabe 2

3. Finde **die Namen** der Schauspieler, die “Fletcher Christian” spielen in einer Produktion des Films “Mutiny on the Bounty” und **das Produktionsjahr** des Films (SQL + rel.A.)

```
SELECT S.Name, F.Jahr
FROM Schauspieler S INNER JOIN Casting C
    ON S.ID = C.SchauspielerID
    INNER JOIN Filme F ON C.FilmId = F.ID
WHERE C.Charakter = 'Fletcher Christian'
    AND F.Titel = 'Mutiny on the Bounty'
```

## Aufgabe 2

### 3. Relationale Algebra:

$\rho_S(\text{Schauspieler})$

$\rho_C(\sigma_{\text{Casting.Character} = \text{'Fletcher Christian'}}(\text{Casting}))$

$\rho_F(\sigma_{\text{Filme.Titel} = \text{'Mutiny on the Bounty'}}(\text{Filme}))$

$\Pi_{S.Name, F.Jahr} (S \bowtie_{S.ID = C.SchauspielerID} C$   
 $\bowtie_{C.FilmID = F.ID} F)$

## Aufgabe 2

4. Finde die Namen der Schauspieler die 'Superman' und 'Clark Kent' gespielt haben (SQL + rel. Alg)

```
SELECT DISTINCT S1.Name
FROM Schauspieler S1 INNER JOIN Casting C1
  ON C1.SchauspielerID = S1.ID
  INNER JOIN Schauspieler S2 ON S1.ID = S2.ID
  INNER JOIN Casting C2
    ON C2.SchauspielerID = S2.ID
WHERE C1.Charakter = 'Superman'
      AND C2.Charakter = 'Clark Kent'
```



# Aufgabe 2

## 4. Relationale Algebra:

$$\begin{aligned} & \Pi_{\text{Schauspieler.Name}} (\text{Schauspieler} \\ & \bowtie_{\text{Schauspieler.ID} = \text{Casting.SchauspielerID}} \\ & \sigma_{\text{Casting.Character} = \text{'Superman'}} (\text{Casting}) ) \\ & \cap \\ & \Pi_{\text{Schauspieler.Name}} (\text{Schauspieler} \\ & \bowtie_{\text{Schauspieler.ID} = \text{Casting.SchauspielerID}} \\ & \sigma_{\text{Casting.Character} = \text{'Clark Kent'}} (\text{Casting}) ) \end{aligned}$$

## Aufgabe 2

5. Finde die Namen der Schauspieler die 'Superman' und 'Clark Kent' gespielt haben in derselben Produktion des Films (SQL)

```
SELECT DISTINCT S.Name
FROM Schauspieler S INNER JOIN Casting C1
    ON C1.SchauspielerID = S.ID
    INNER JOIN Casting C2
    ON C2.SchauspielerID = S.ID
WHERE C1.Charakter = 'Superman'
    AND C2.Charakter = 'Clark Kent'
    AND C1.FilmID = C2.FilmID
```

## Aufgabe 2

6. Finde die Namen der Schauspieler die zwei unterschiedliche Charaktern in derselben Filmproduktion gespielt haben (SQL)

```
SELECT DISTINCT S.Name
FROM Schauspieler S INNER JOIN Casting C1
  ON C1.SchauspielerID = S.ID
  INNER JOIN Casting C2
    ON C2.SchauspielerID = S.ID
WHERE C1.Charakter <> C2.Charakter
      AND C1.FilmId = C2.FilmID
```

## Aufgabe 2

7. Finde die Paare von unterschiedlichen Schauspielernamen, die denselben Charakter in unterschiedlichen Filmproduktionen desselben Films gespielt haben (SQL)

```
SELECT S1.Name, S2.Name
FROM Schauspieler S1 INNER JOIN Casting C1
  ON C1.SchauspielerID = S1.ID
INNER JOIN Filme F1 ON C1.FilmID = F1.ID,
Schauspieler S2 INNER JOIN Casting C2
  ON C2.SchauspielerID = S2.ID
INNER JOIN Filme F2 ON C2.FilmID = F2.ID
WHERE F1.Titel = F2.Titel
      AND F1.Jahr <> F2.Jahr
      AND C1.Charakter = C2.Charakter
      AND S1.ID <> S2.ID
```

## Aufgabe 2

8. Finde die Namen der Schauspieler, die zusammen mit Victoria Abril in einem Film gespielt haben und die geboren wurden, nachdem Victoria in dem ersten Film gespielt hat (SQL)

```
SELECT S1.Name FROM Schauspieler S1 INNER JOIN Casting C1
    ON S1.ID = C1.SchauspielerID ,
    Schauspieler S2 INNER JOIN Casting C2
    ON S2.ID = C2.SchauspielerID
WHERE S2.Name = 'Victoria Abril' AND
    C2.FilmId = C1.FilmID AND S1.ID <> S2.ID
AND S1.Geburtsjahr >
    (SELECT MIN(F.Jahr) FROM Filme F
        INNER JOIN Casting C ON C.FilmID = F.ID
        INNER JOIN Schauspieler S ON C.SchauspielerID = S.ID
        WHERE S.Name = 'Victoria Abril')
```

## Aufgabe 2

9. Finde die Namen der Schauspieler die 'Superman' oder 'Clark Kent' gespielt haben. (SQL + rel. Alg.)

```
SELECT DISTINCT S.Name
FROM Schauspieler S, Casting C
WHERE C.SchauspielerID = S.ID
      AND (C.Charakter = 'Superman'
      OR C.Charakter = 'Clark Kent')
```

# Aufgabe 2

## 9. Relationale Algebra:

$\Pi_{\text{Schauspieler.Name}} ((\text{Schauspieler} \bowtie$

$\sigma_{\text{Casting.Charakter} = \text{'Superman'} \vee \text{Casting.Character} = \text{'Clark Kent'}} (\text{Casting}))$

# Aufgabe 3

Die Datenbank enthält folgende Relationen:

- Lieferanten (LID, Lname, Adresse)
- Produkte (PID, Pname, Farbe)
- Katalog (LID, PID, Preis)

Die Primärschlüssel sind unterstrichen.

Katalog enthält die Preise für Produkte von unterschiedlichen Lieferanten.

Geben sie an, was die folgenden Anfragen ausgeben.



## Aufgabe 3

1.  $\pi_{\text{LName}}(\pi_{\text{LID}}((\sigma_{\text{Farbe} = \text{'rot'}} \text{Produkte}) \bowtie (\sigma_{\text{Preis} < 100} \text{Katalog})) \bowtie \text{Lieferanten})$

Finde die Lieferantennamen, die ein rotes Produkt liefern, das weniger als 100 kostet.

2.  $\pi_{\text{LName}}(\pi_{\text{LID}}((\sigma_{\text{Farbe} = \text{'rot'}} \text{Produkte}) \bowtie (\sigma_{\text{Preis} < 100} \text{Katalog}) \bowtie \text{Lieferanten}))$

Diese Anfrage gibt nichts aus. Nachdem wir auf das Attribut SID projizieren, ist das das einzige Attribut in der Menge. Dann können wir nicht mehr auf Lname projizieren.

## Aufgabe 3

$$3. \quad \pi_{\text{LName}}((\sigma_{\text{Farbe} = \text{'rot'}} \text{Produkte}) \bowtie (\sigma_{\text{Preis} < 100} \text{Katalog}) \bowtie \text{Lieferanten}) \cap \\ \pi_{\text{LName}}((\sigma_{\text{Farbe} = \text{'grün'}} \text{Produkte}) \bowtie (\sigma_{\text{Preis} < 100} \text{Katalog}) \bowtie \text{Lieferanten})$$

Finde die Lieferantennamen, die ein rotes Produkt liefern, das weniger als 100 kostet und ein grünes Produkt, das weniger als 100 kostet.

$$4. \quad \pi_{\text{LName}}((\sigma_{\text{Farbe} = \text{'rot'}} \text{Produkte}) \bowtie (\sigma_{\text{Preis} < 100} \text{Katalog}) \bowtie \text{Lieferanten}) \cup \\ \pi_{\text{LName}}((\sigma_{\text{Farbe} = \text{'grün'}} \text{Produkte}) \bowtie (\sigma_{\text{Preis} < 100} \text{Katalog}) \bowtie \text{Lieferanten})$$

Finde die Lieferantennamen, die ein rotes oder grünes Produkt liefern, das weniger als 100 kostet.

## Aufgabe 3

5.  $(\pi_{LID, PID} \text{ Katalog}) \div \pi_{PID} \text{ Produkte}$

$R_1 \div R_2$  enthält alle  $x$ , sodass für jeder  $y$  in  $R_2$ , es gibt ein  $xy$  in  $R_1$ .  
Finde alle LIDs der Lieferanten die jeder Produkt liefern.

6.  $\rho_{R1}(\text{Katalog})$  ,  $\rho_{R2}(\text{Katalog})$

$$\pi_{R1.PID}(\sigma_{R1.PID = R2.PID \wedge R1.LID \neq R2.LID} (R1 \times R2))$$

Finde die PIDs der Produkte, die von wenigstens zwei Lieferanten geliefert werden.

## Aufgabe 4

Um die Bücher in der Bibliothek zu speichern brauchen wir:

*BuchID, ISBN, Kategorie, SerialNr, KopieNr, Titel, Author*

Jedes Buch (jede Kopie) hat eine eindeutige Nummer (BuchID) den wir benutzen können um die Anleihe des Buches zu speichern. Diese Nummer ist einem Buch zugeordnet als ein Barcode, hat aber keine besondere Bedeutung.

ISBN ist eindeutig für eine Edition eines Buches.

Jedes Buch gehört zu einer Kategorie. In einer Kategorie haben die Bücher unterschiedliche SerialNr. Wenn wir mehrere Kopien von demselben Buch haben, dann werden diese durch den KopieNr unterschiedet.

Welche der folgenden fkt. Abh. gelten? (Nicht aus dem Text, sonder logisch in dem beschriebenen Kontext) Gebe eine kurze Erklärung dafür.

## Aufgabe 4

1. BuchID  $\rightarrow$  ISBN, Kategorie, SerialNr, KopieNr

**Ja:** jedes Buch hat ein eindeutiges BuchId (BuchID ist ein Kandidatschlüssel)

2. ISBN  $\rightarrow$  BuchID

**Nein:** es kann mehrere Kopien von demselben Buch geben mit demselben ISBN

3. ISBN  $\rightarrow$  Kategorie, SerialNr

**Ja:** diese Bedingung erzwingt den Bibliothekar konsistent zu sein

# Aufgabe 4

4. ISBN  $\rightarrow$  KopieNr

**Nein:** : es kann mehrere Kopien von demselben Buch geben mit demselben ISBN

5. ISBN  $\rightarrow$  Titel, Author

**Ja:** Ein ISBN ist eindeutig für ein Buchedition, also es gibt nicht unterschiedliche Titel und Autoren für denselben ISBN

6. KopieNr  $\rightarrow$  ISBN, BuchID

**Nein:** die KopieNr kann nicht eindeutig das Buch identifizieren

# Aufgabe 5

Die Ausprägung einer Relation  $S(A, B, C)$  enthält folgende Tupeln:

(1,2,3)

(4,2,3)

(5,3,3)

1. Welche der folgenden fkt. Abh. können für die Relation  $S$  nicht gelten?

a)  $A \rightarrow B$

b)  $BC \rightarrow A$

c)  $B \rightarrow C$

2. Könnt ihr fkt. Abh. identifizieren, die für die Relation  $S$  gelten?

## Aufgabe 5 - Lösung

1. Welche der folgenden fkt. Abh. können für die Relation S nicht gelten?

a)  $A \rightarrow B$  –

b)  $BC \rightarrow A$  - kann nicht gelten wegen der Tupeln (1,2,3) und (4,2,3) :  
unterschiedliche Werte für A mit denselben Werten für BC

c)  $B \rightarrow C$  –

2. Könnt ihr fkt. Abh. identifizieren, die für die Relation S gelten?

**Nein.** Wenn wir sagen, dass die fkt. Abh. für die ganze Relation gilt, so muss diese für alle gültige Ausprägungen gelten.



## Aufgabe 6

In der folgenden Relation speichern wir die Betreuer für jeden Student. Ein Student kann aber gleichzeitig mehrere Hauptfächer studieren (um am Ende mehrere Diplomen zu kriegen, z.B. Mathematik und Informatik). Dafür muss er aber auch mehrere Diplomarbeit schreiben.

(StudentID, Hauptfach, Betreuer)

1. Welche sind die Kandidatschlüssel?
2. Ist die Relation in 1NF, 2NF, 3NF, BCNF? Erkläre

# Aufgabe 6 - Lösung

(StudentID, Hauptfach, Betreuer)

1. Kandidatschlüssel: (StudentID, Hauptfach), (StudentID, Betreuer)
2. Ist die Relation in 1NF, 2NF, 3NF, BCNF?

Folgende fkt. Abh gelten:

- StudentID, Hauptfach  $\rightarrow$  Betreuer
  - StudentID, Betreuer  $\rightarrow$  Hauptfach
  - Betreuer  $\rightarrow$  Hauptfach ...
- a) 1NF – Ja (alle Werte in der Relation sind atomar)
  - b) 2NF – Ja (es gibt keine Nichtschlüsselattribute)
  - c) 3NF – Ja (Betreuer  $\rightarrow$  Hauptfach , Hauptfach ist prim)
  - d) BCNF – Nein (Betreuer  $\rightarrow$  Hauptfach verletzt BCNF)

# Aufgabe 7

Sei die Schema  $R(A, B, C, D, E)$  mit folgenden fkt. Abh.:

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

1. Finde alle Kandidatschlüssel der Relation  $R$ .
2. Berechne die kanonische Überdeckung von  $F$ .
3. Ist  $R$  in BCNF? Erkläre.
4. Finde eine verlustlose BCNF Zerlegung von  $R$ .
5. Ist die Zerlegung von Punkt 4. abhängigkeitsbewahrend? Erkläre.
6. Ist  $R$  in 3NF? Erkläre.
7. Berechne mithilfe des Synthesealgorithmus eine 3NF Zerlegung von  $R$ .

# Aufgabe 7 - Lösung

1. Finde alle Kandidatschlüssel der Relation R.

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

$$A^+ = A, B^+ = BCDE, C^+ = BCDE, D^+ = D$$

Wir merken, dass kein Attribut A bestimmt  $\Rightarrow$  A gehört zu dem Schlüssel

$$AB^+ = ABCDE \Rightarrow \text{Kandidatschlüssel}$$

$$AC^+ = ABCDE \Rightarrow \text{Kandidatschlüssel}$$

$$AD^+ = AD$$

$$AE^+ = AE$$

$$ADE^+ = ADE$$

# Aufgabe 7 - Lösung

## 2. Berechne die kanonische Überdeckung von F.

- Schritt 1 : Linksreduktion:  $A \rightarrow B$ ,  $X \in A$ , falls  $B \subset (A - \{X\})^+$  bzgl.  $F \Rightarrow$  reduziere  $X$  (ersetze  $A \rightarrow B$  durch  $A - \{X\} \rightarrow B$ )
- Schritt 2 : Rechtsreduktion:  $A \rightarrow B$ ,  $Y \in B$ , falls  $Y \in A^+$  bzgl.  $F - (A \rightarrow B) \cup (A \rightarrow B - \{Y\}) \Rightarrow$  reduziere  $Y$  (ersetze  $A \rightarrow B$  durch  $A \rightarrow B - \{Y\}$ )
- Schritt 3 : Entferne die FDs der Form  $A \rightarrow \emptyset$
- Schritt 4 : Ersetze alle FDs der Form  $A \rightarrow B_1, \dots, A \rightarrow B_k$  durch  $A \rightarrow B_1 \cup \dots \cup B_k$

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

Schritt 1.  $\{B \rightarrow CDE, C \rightarrow BDE, B \rightarrow C, C \rightarrow B, C \rightarrow D, B \rightarrow E\}$

Schritt 2.  $\{B \rightarrow \emptyset, C \rightarrow \emptyset, B \rightarrow C, C \rightarrow B, C \rightarrow D, B \rightarrow E\}$

Schritt 3.  $\{B \rightarrow C, C \rightarrow B, C \rightarrow D, B \rightarrow E\}$

Schritt 4.  $F_c = \{B \rightarrow CE, C \rightarrow BD\}$

# Aufgabe 7 - Lösung

3. Ist R in BCNF? Erkläre.

R ist in BCNF wenn für alle Abhängigkeiten  $A \rightarrow B$  aus  $F^+$  gilt:

- $B \subseteq A$  (FD ist trivial) **oder**
- A enthält einen Schlüssel von R (A ist ein Superschlüssel)

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

Kandidatschlüssel AB, AC

$B \rightarrow C$  verletzt BCNF  $\Rightarrow$  R nicht in BCNF

# Aufgabe 7 - Lösung

4. Finde eine verlustlose BCNF Zerlegung von R.

- Wenn die  $\alpha \rightarrow \beta$  die BCNF verletzt, dann können wir die Relation in  $R - \beta$  und  $\alpha \cup \beta$  zerlegen.

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

- $B \rightarrow C$  verletzt BCNF

Zerlege R in  $\{\underline{ABDE}\}$  (zugeordnete FDs  $B \rightarrow E$ ) und  $\{\underline{BC}\}$  (zugeordnete FDs  $B \rightarrow C, C \rightarrow B$ )

- $B \rightarrow E$  verletzt BCNF

Zerlege R in  $\{\underline{ABD}\}, \{\underline{BE}\}$  und  $\{\underline{BC}\} \Rightarrow$  BCNF Zerlegung

## Aufgabe 7 - Lösung

5. Ist die Zerlegung von Punkt 4. abhängigkeitsbewahrend? Erkläre.

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

$F_c = \{B \rightarrow CE, C \rightarrow BD\}$

Zerlegung:  $\{\underline{ABD}\}, \{\underline{BE}\}$  (zugeordnete FDs  $B \rightarrow E$ ) und  $\{\underline{BC}\}$  }  
(zugeordnete FDs  $B \rightarrow C, C \rightarrow B$ )

Diese Zerlegung ist verlustlos, aber nicht abhängigkeitsbewahrend ( $C \rightarrow D$  ist nicht lokal überprüfbar)



# Aufgabe 7 - Lösung

6. Ist R in 3NF? Erkläre.

R ist in 3NF wenn für alle Abhängigkeiten  $A \rightarrow B$  aus  $F^+$  gilt:

- $B \subseteq A$  (FD ist trivial) **oder**
- A enthält einen Schlüssel von R (A ist ein Superschlüssel) **oder**
- B ist Teil eines Schlüsselkandidaten (B ist prim)

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

Kandidatschlüssel AB, AC

$C \rightarrow D$  verletzt 3NF  $\Rightarrow$  R nicht in 3NF

# Aufgabe 7 - Lösung

7. Berechne mithilfe des Synthesealgorithmus eine 3NF Zerlegung von R.

Synthesealgorithmus:

1. Bestimme die kanonische Überdeckung  $F_c$  der Menge F
2. Führe für jede FD  $A \rightarrow B$  in  $F_c$  folgende Anweisungen:  
Erzeuge eine Relation  $R_A = A \cup B$  und ordne  $R_A$  die FDs  $F_A = \{C \rightarrow D \in F_c \mid C \cup D \subseteq R_A\}$  zu
3. Falls alle Relationen erzeugt in Schritt 2 keinen Schlüsselkandidaten des ursprünglichen Relation R enthalten, so erzeuge zusätzlich eine neue Relation  $R_K = K$  und  $F_K = \emptyset$ , wobei K ein Schlüsselkandidat von R ist
4. Eliminiere die Relationen  $R_A$ , die in einem anderen Schema enthalten sind, d.h.  $R_i \subseteq R_j$

# Aufgabe 7 - Lösung

7. Berechne mithilfe des Synthesealgorithmus eine 3NF Zerlegung von R.

Schritt 1.  $F_C = \{B \rightarrow CE, C \rightarrow BD\}$  – kanonische Überdeckung

Schritt 2.  $R_1 = \{BCE\}$  (zugeordnete FDs  $B \rightarrow CE$ ) ,

$R_2 = \{BCD\}$  (zugeordnete FDs  $C \rightarrow BD$ )

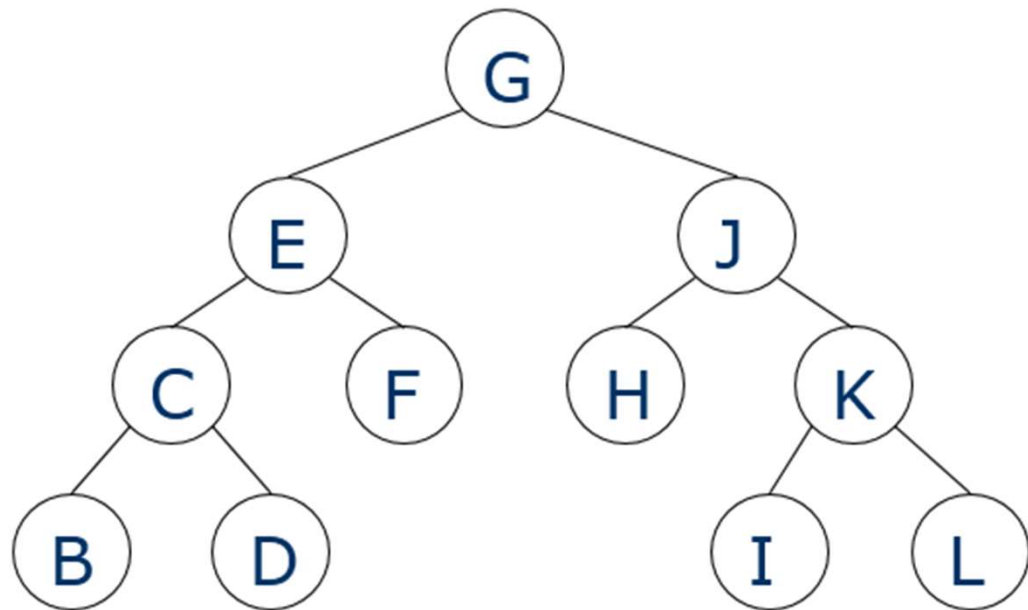
Schritt 3.  $R_3 = \{AB\}$  ( keine zugeordnete FDs)

Schritt 4. –

$\Rightarrow \{BCE\}, \{BCD\}, \{AB\}$  – 3NF Zerlegung

## Aufgabe 8

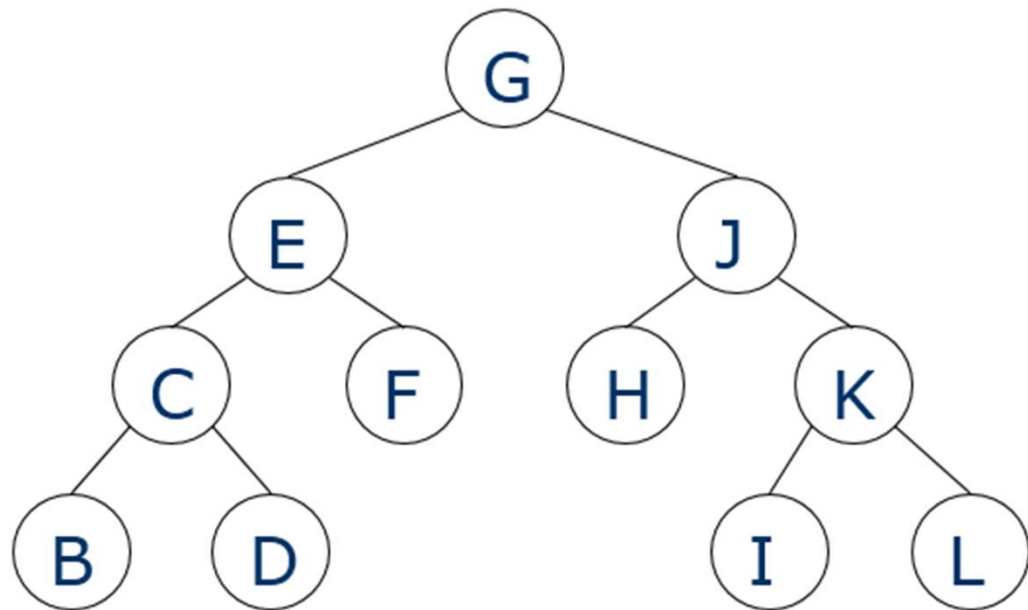
Gegeben sei der folgende balancierte Binärbaum. Beschreibe die Operationen, die ausgeführt werden müssen beim Einfügen des Wertes A, damit der Baum balanciert bleibt.



# Aufgabe 8

Gegeben sei der folgende balancierte Binärbaum. Beschreibe die Operationen, die ausgeführt werden müssen beim Einfügen des Wertes A, damit der Baum balanciert bleibt.

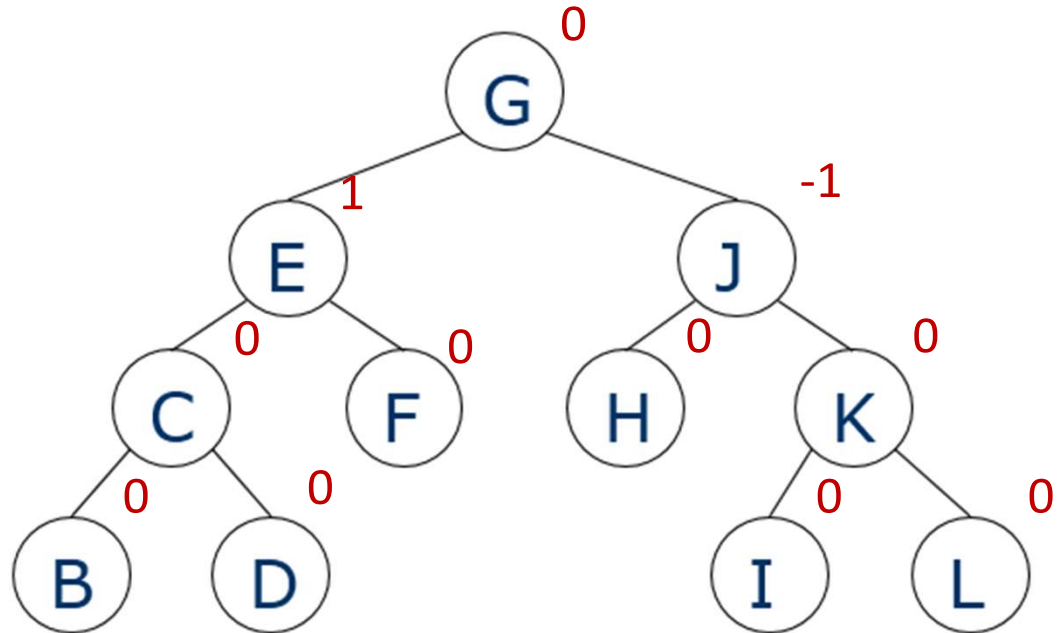
Balancierter Binärbaum =  
die Höhen der zwei Teilbäume  
dürfen mit höchstens 1  
voneinander abweichen



# Aufgabe 8

Gegeben sei der folgende balancierte Binärbaum. Beschreibe die Operationen, die ausgeführt werden müssen beim Einfügen des Wertes A, damit der Baum balanciert bleibt.

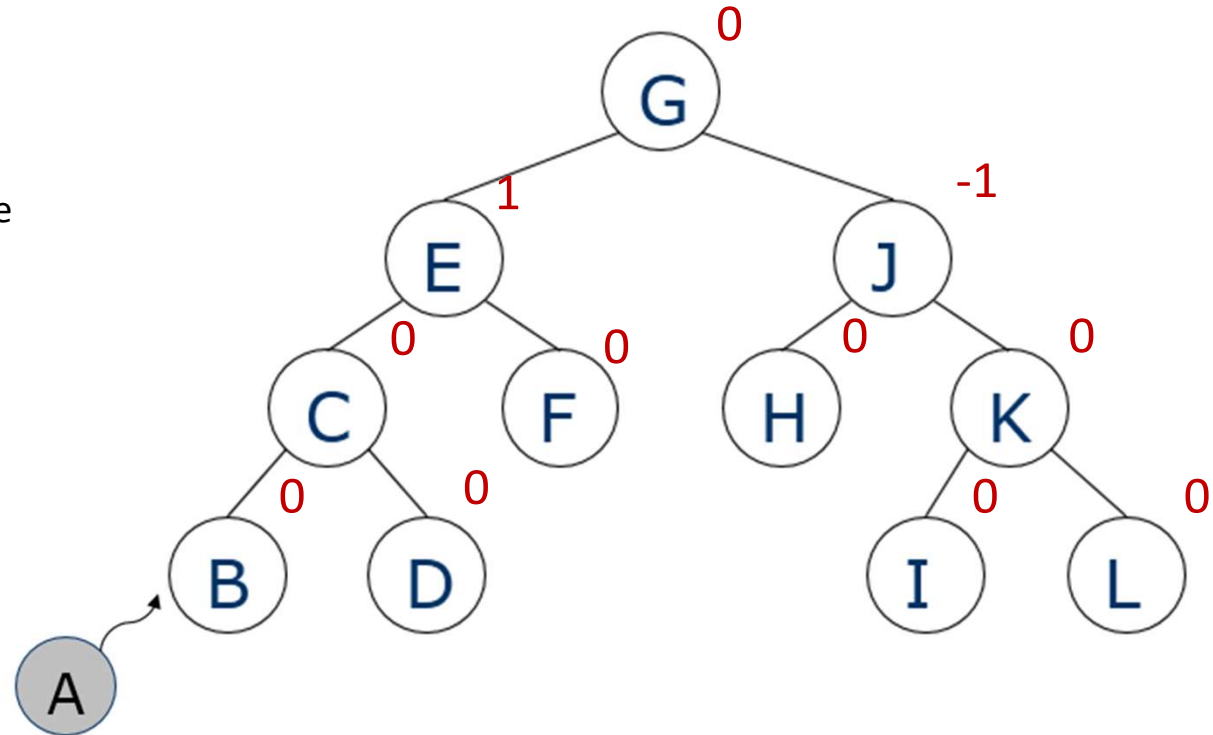
Balancierter Binärbaum =  
die Höhen der zwei Teilbäume  
dürfen mit höchstens 1  
voneinander abweichen



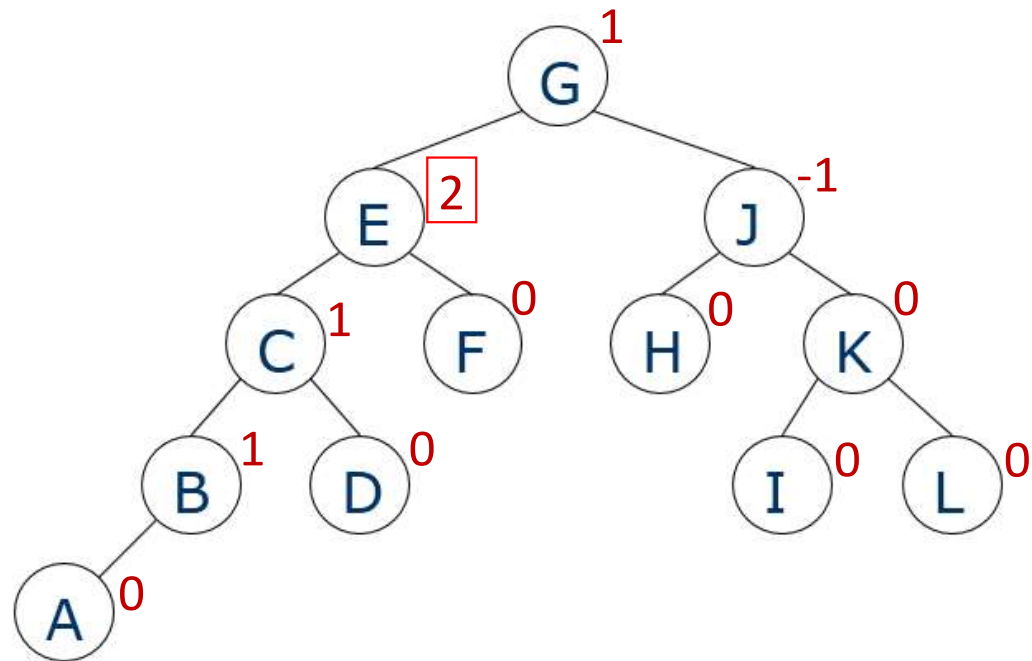
# Aufgabe 8

Gegeben sei der folgende balancierte Binärbaum. Beschreibe die Operationen, die ausgeführt werden müssen beim Einfügen des Wertes A, damit der Baum balanciert bleibt.

Balancierter Binärbaum =  
die Höhen der zwei Teilbäume  
dürfen mit höchstens 1  
voneinander abweichen

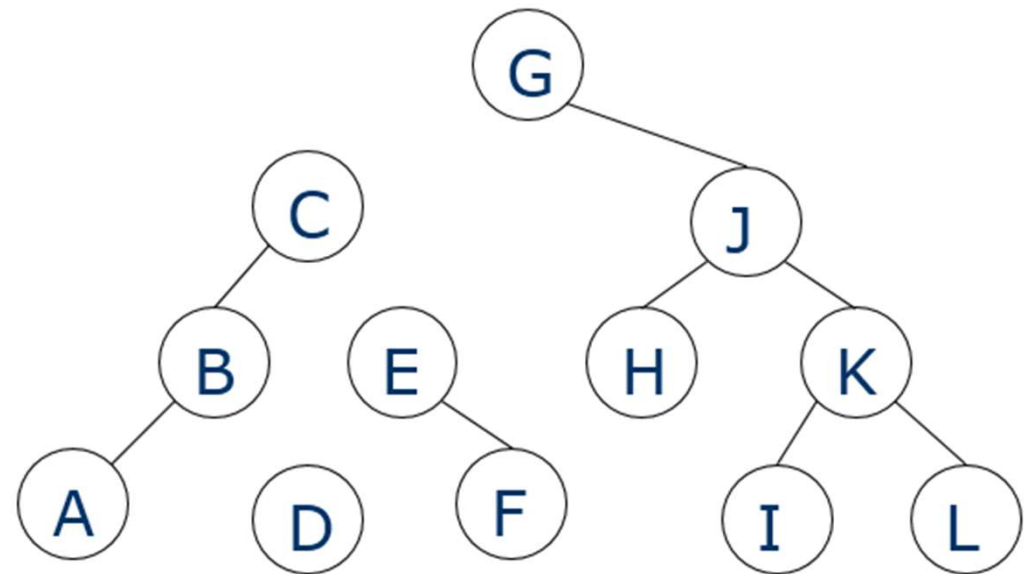


## Aufgabe 8



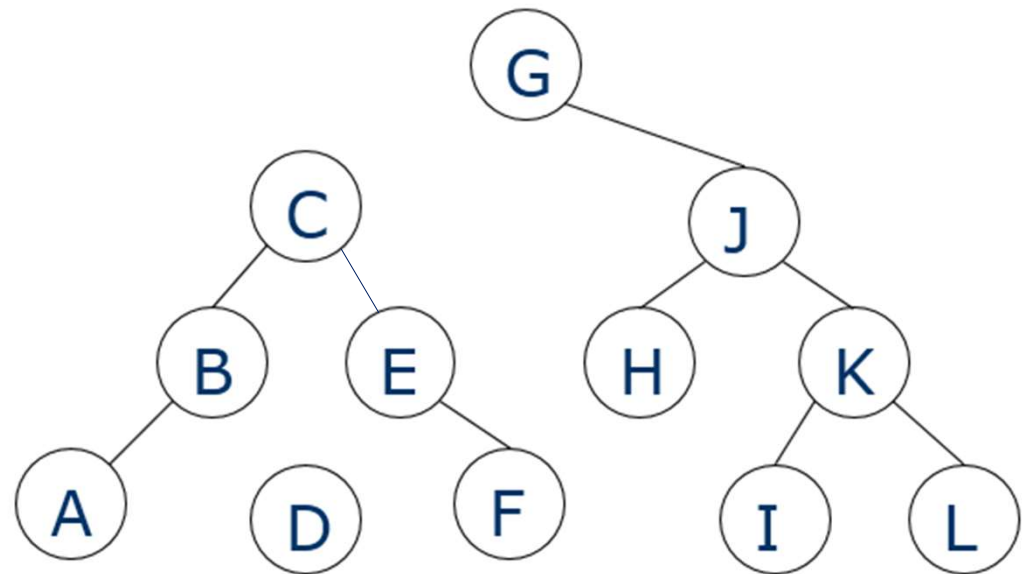


## Aufgabe 8



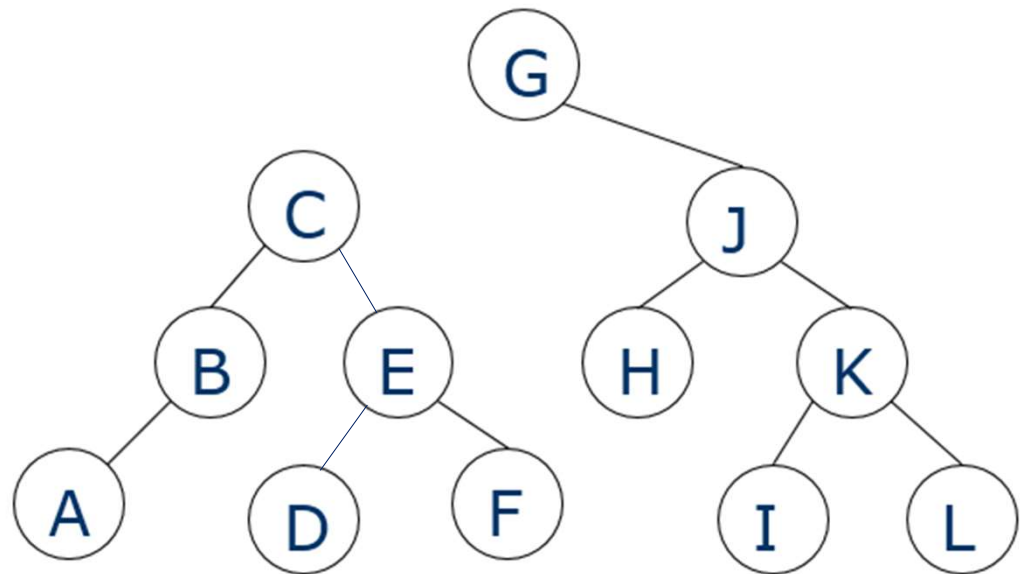
## Aufgabe 8

- Der Knoten *E* wird das rechte Kind des Knotens *C*



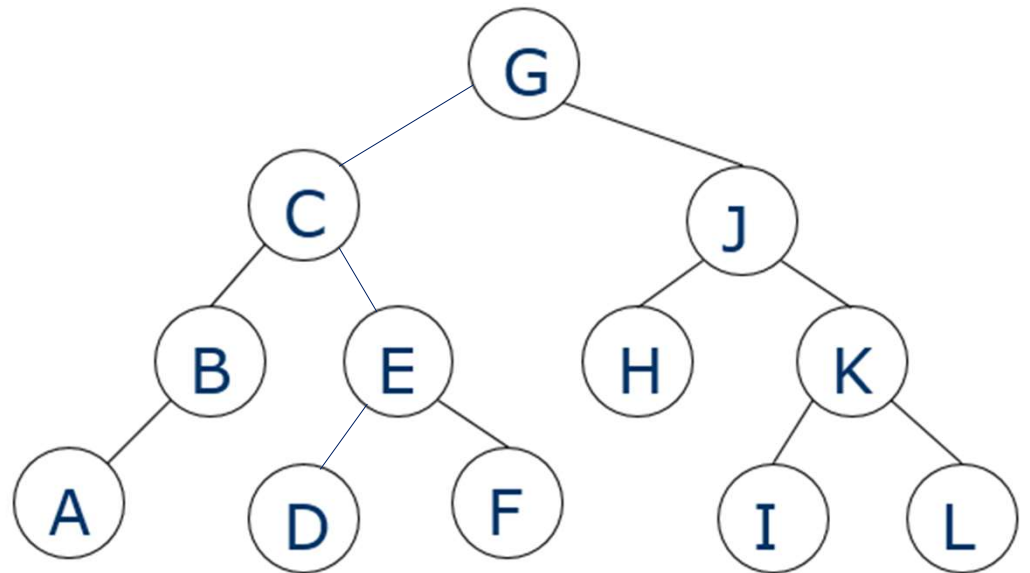
## Aufgabe 8

- Der Knoten *E* wird das rechte Kind des Knotens *C*
- Der Knoten *D* wird das linke Kind des Knotens *E*



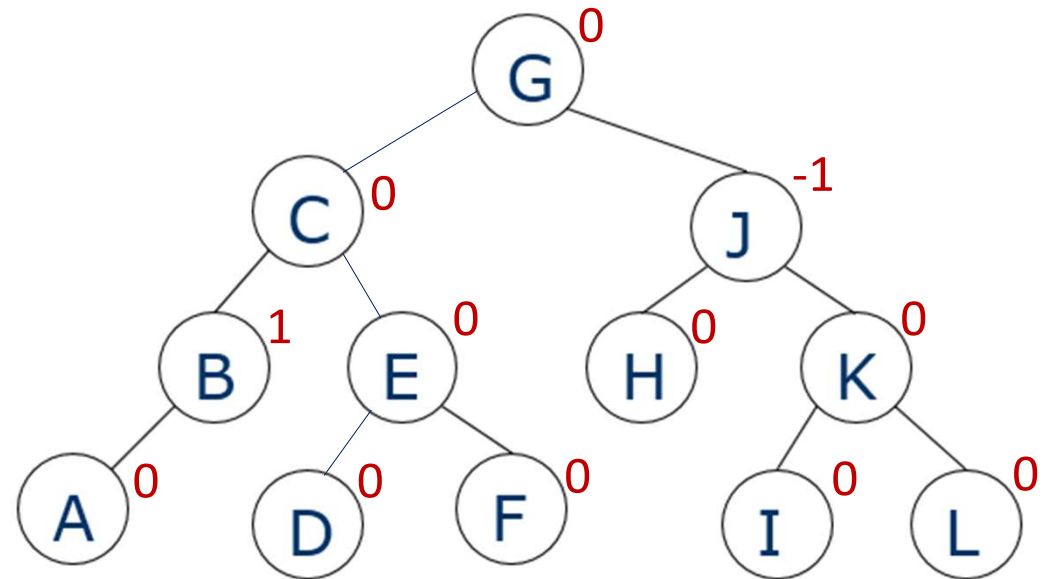
## Aufgabe 8

- Der Knoten *E* wird das rechte Kind des Knotens *C*
- Der Knoten *D* wird das linke Kind des Knotens *E*
- Der Knoten *C* wird das linke Kind des Knotens *G*



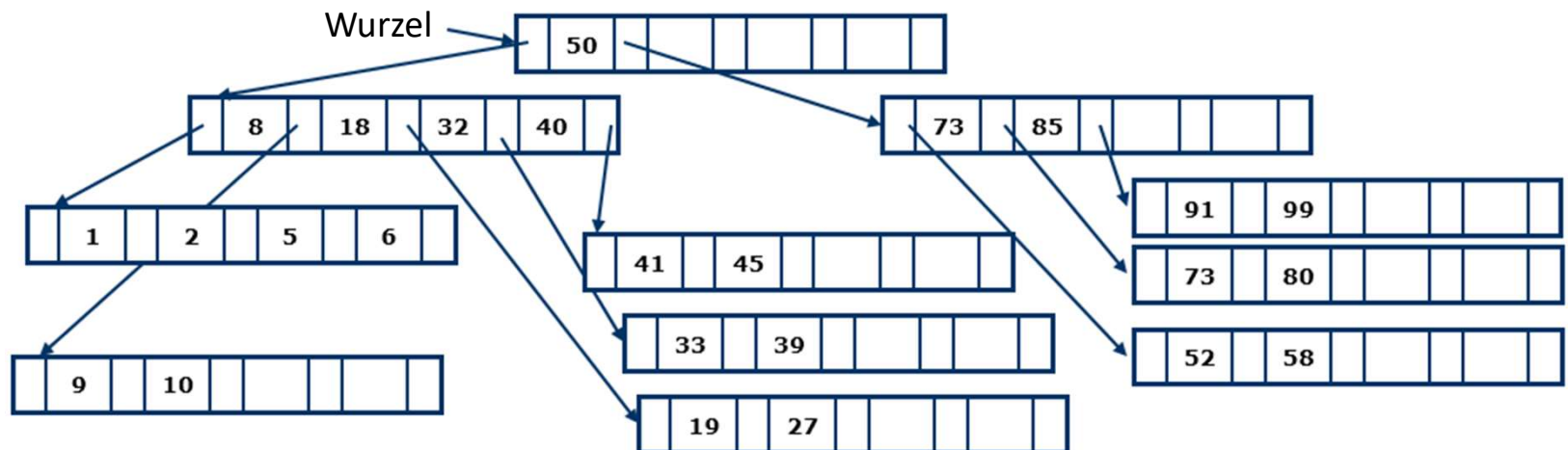
## Aufgabe 8

- Der Knoten *E* wird das rechte Kind des Knotens *C*
- Der Knoten *D* wird das linke Kind des Knotens *E*
- Der Knoten *C* wird das linke Kind des Knotens *G*



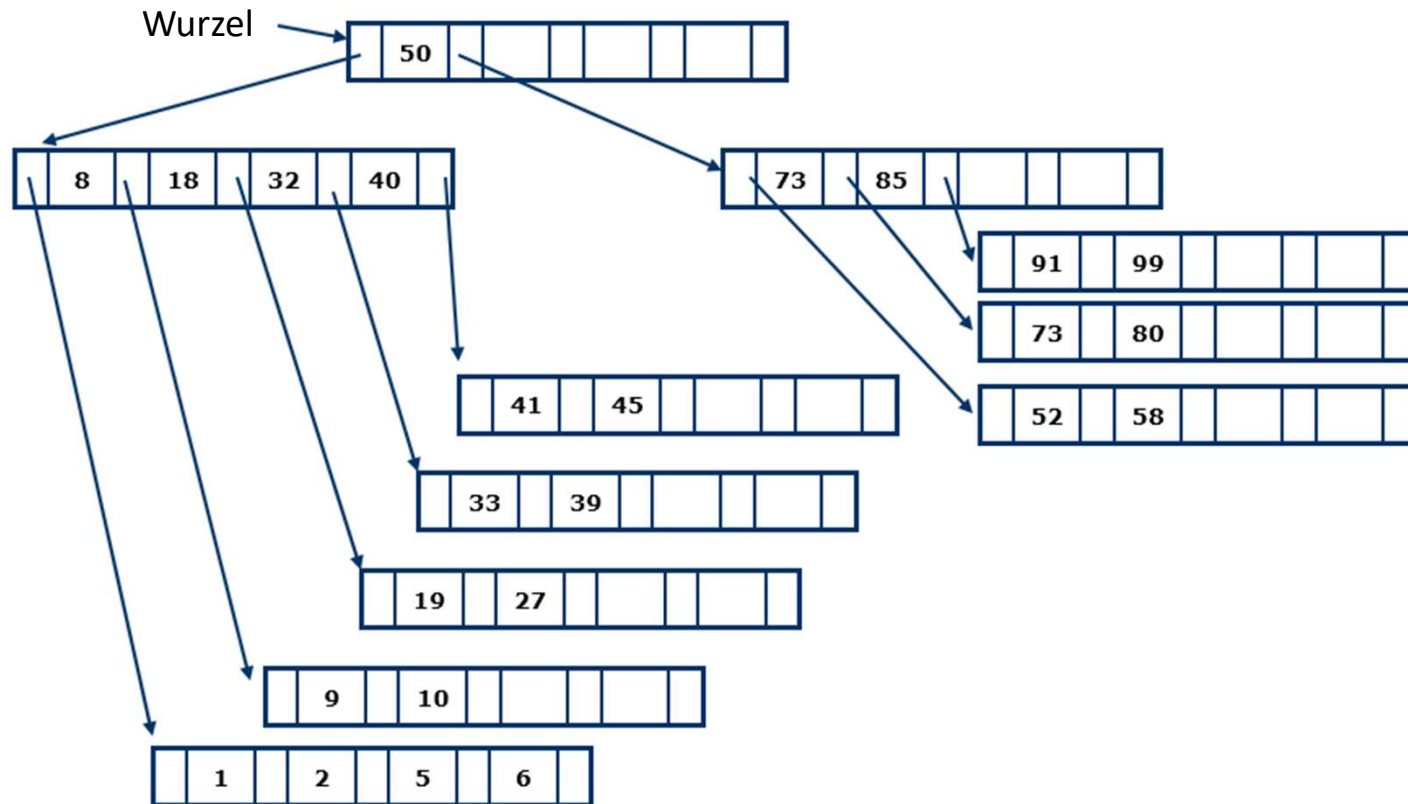
# Aufgabe 9

Gegeben wird folgender B-Baum Index mit Ordnung 2:

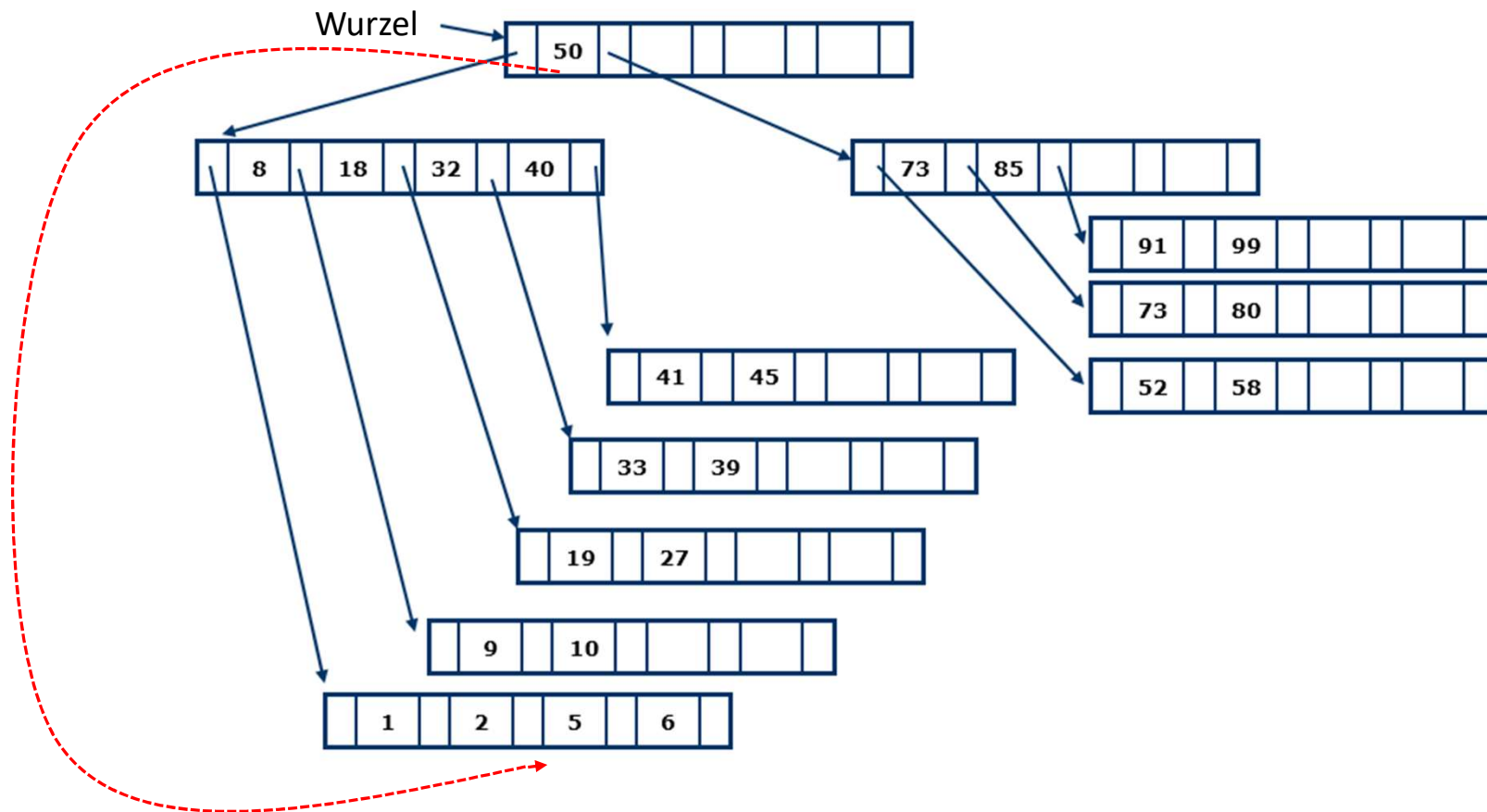


1. Füge den Dateneintrag mit Schlüsselwert „3“ ein.
2. Lösche den Dateneintrag mit Schlüsselwert „8“ mit Balance-Operationen
3. Lösche den Dateneintrag mit Schlüsselwert „8“ mit Merge-Operationen

## Aufgabe 9 – füge „3“ ein

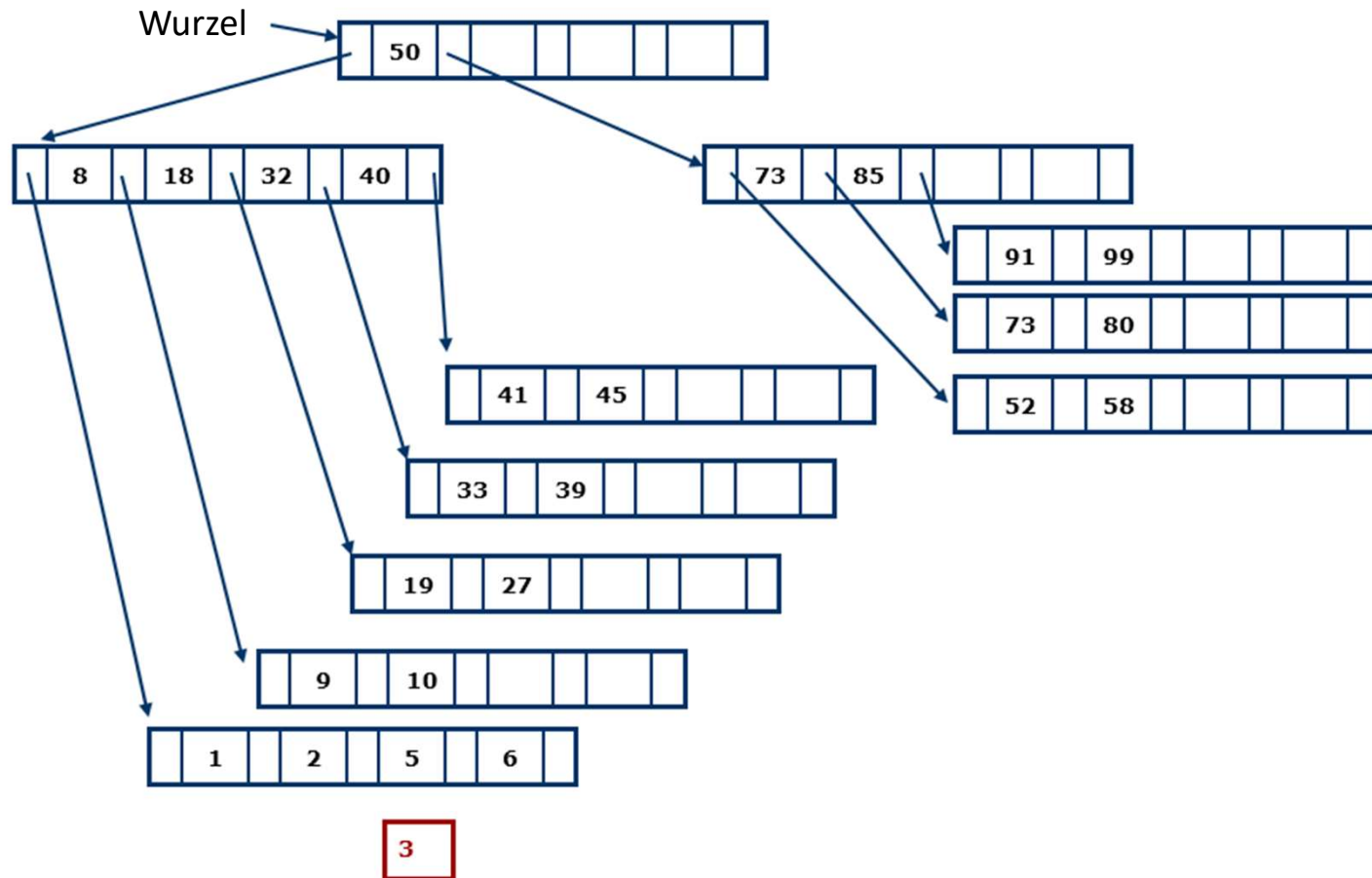


## Aufgabe 9 – füge „3“ ein

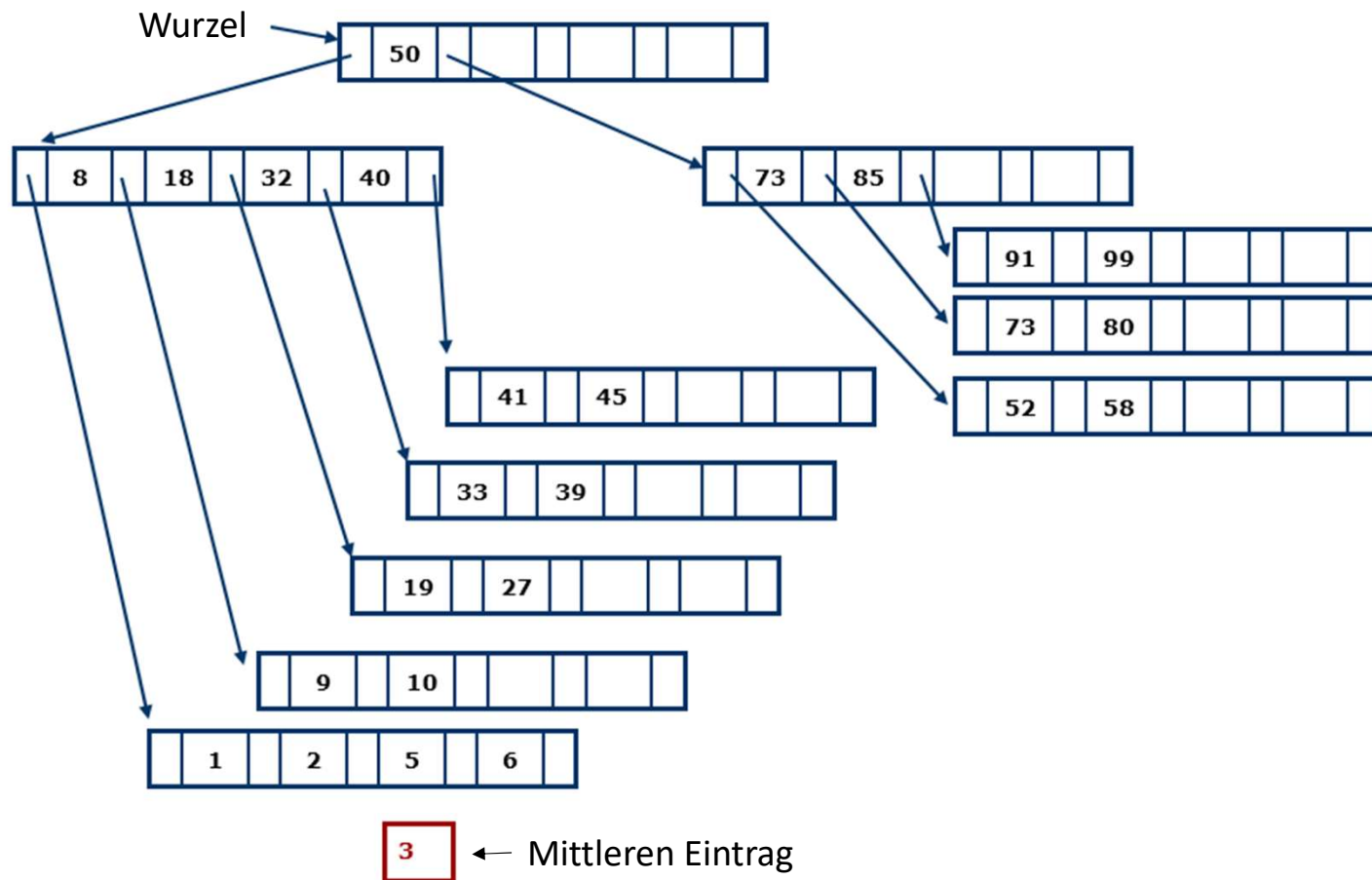




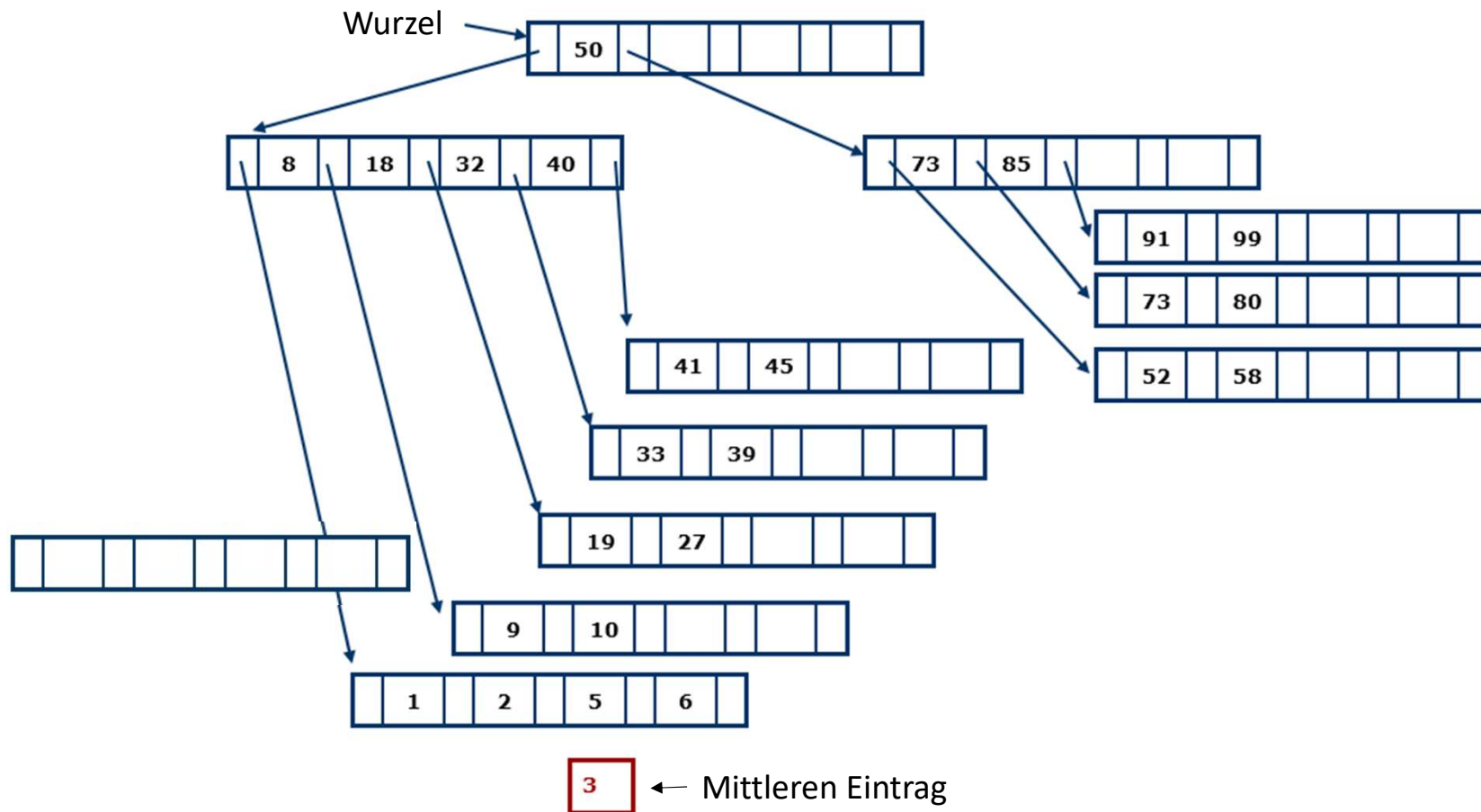
## Aufgabe 9 – füge „3“ ein



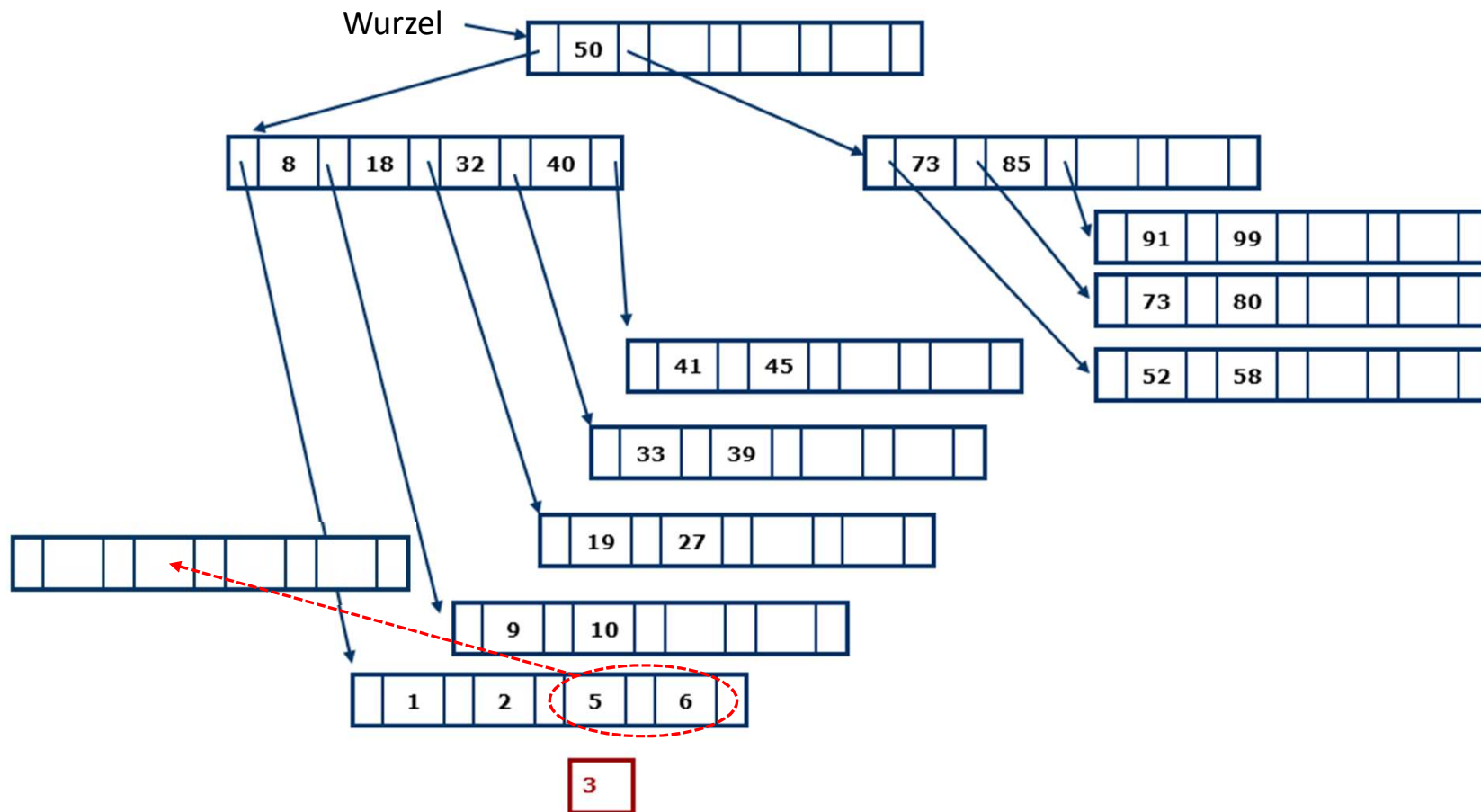
## Aufgabe 9 – füge „3“ ein



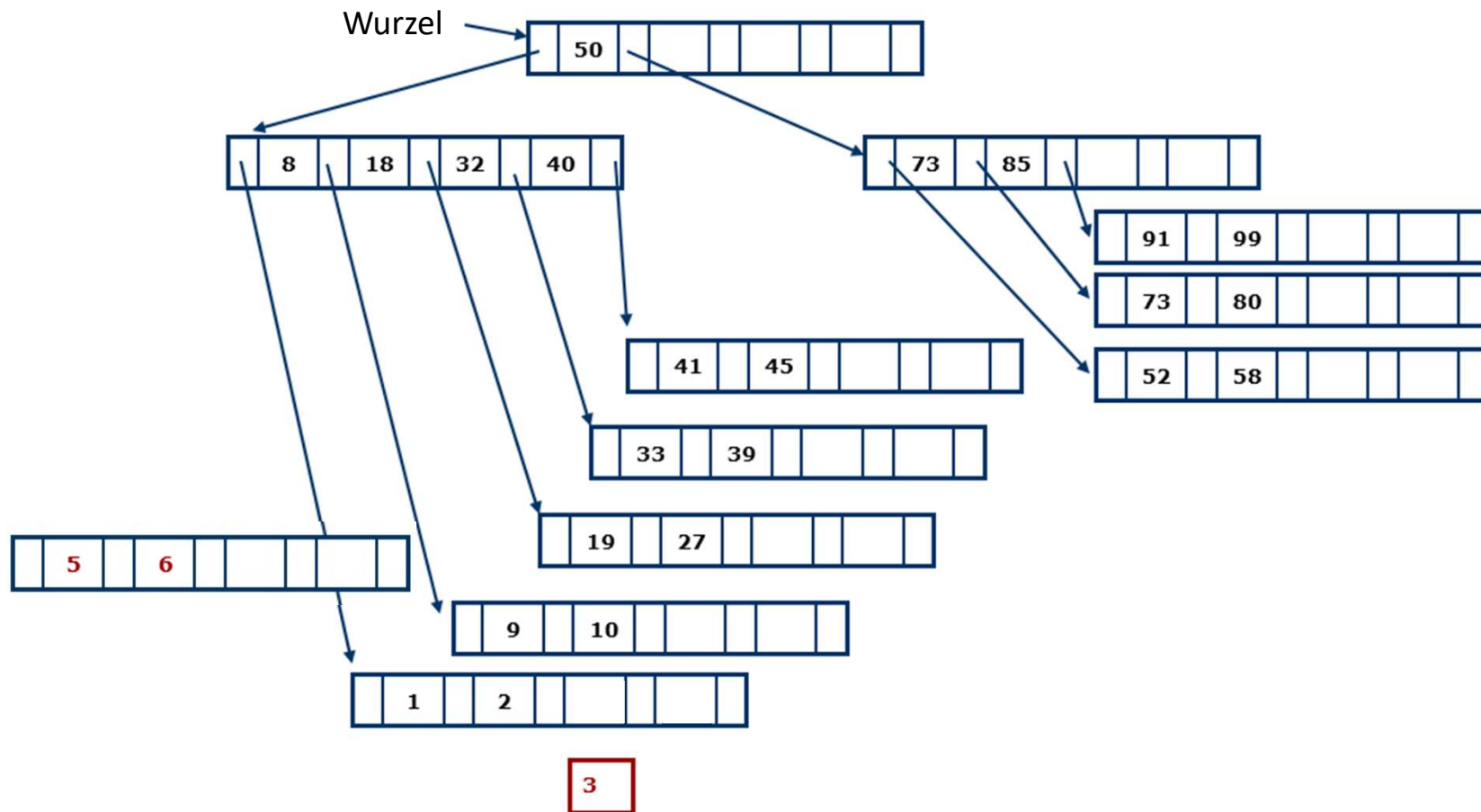
## Aufgabe 9 – füge „3“ ein



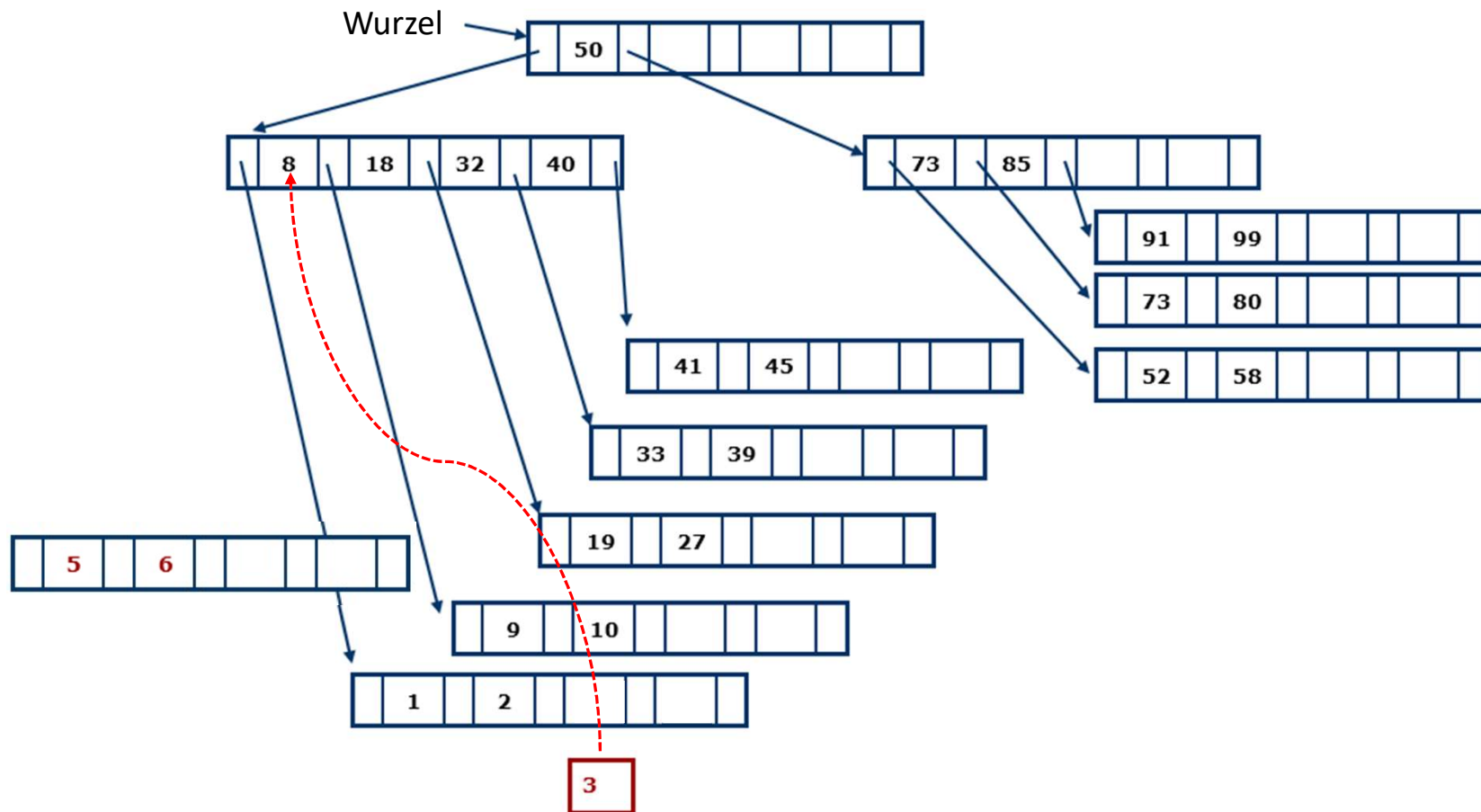
## Aufgabe 9 – füge „3“ ein



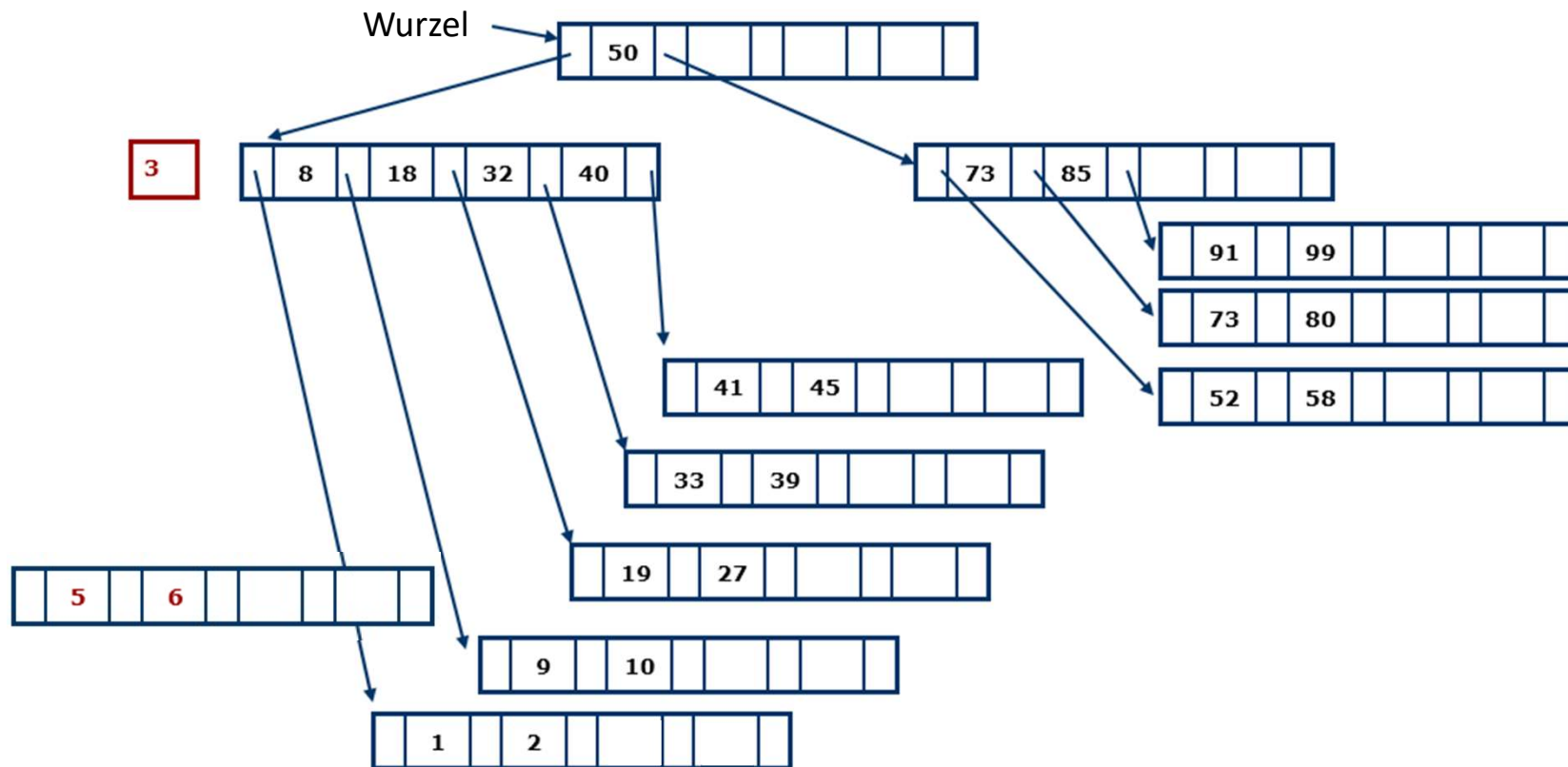
## Aufgabe 9 – füge „3“ ein



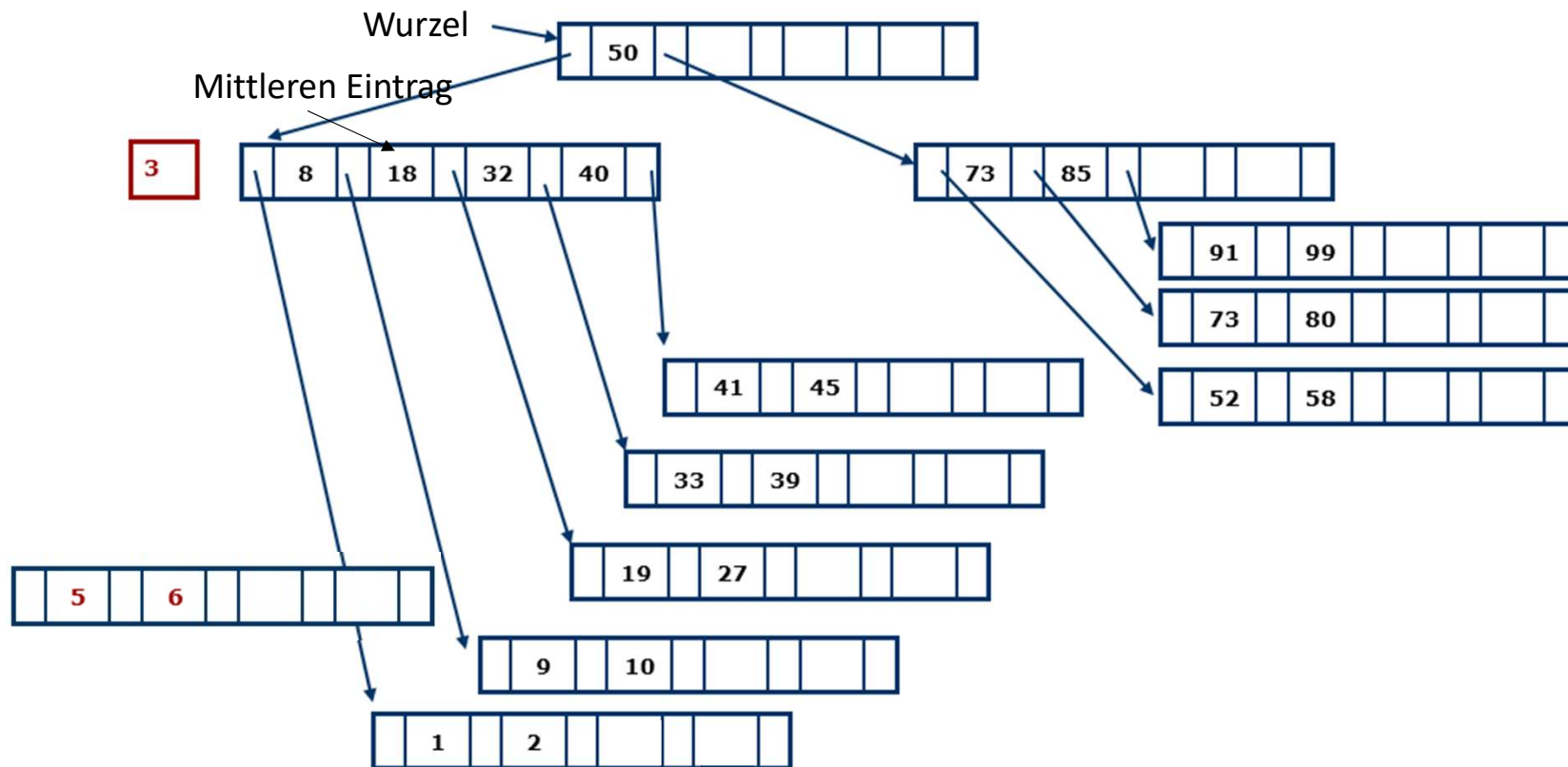
## Aufgabe 9 – füge „3“ ein



## Aufgabe 9 – füge „3“ ein

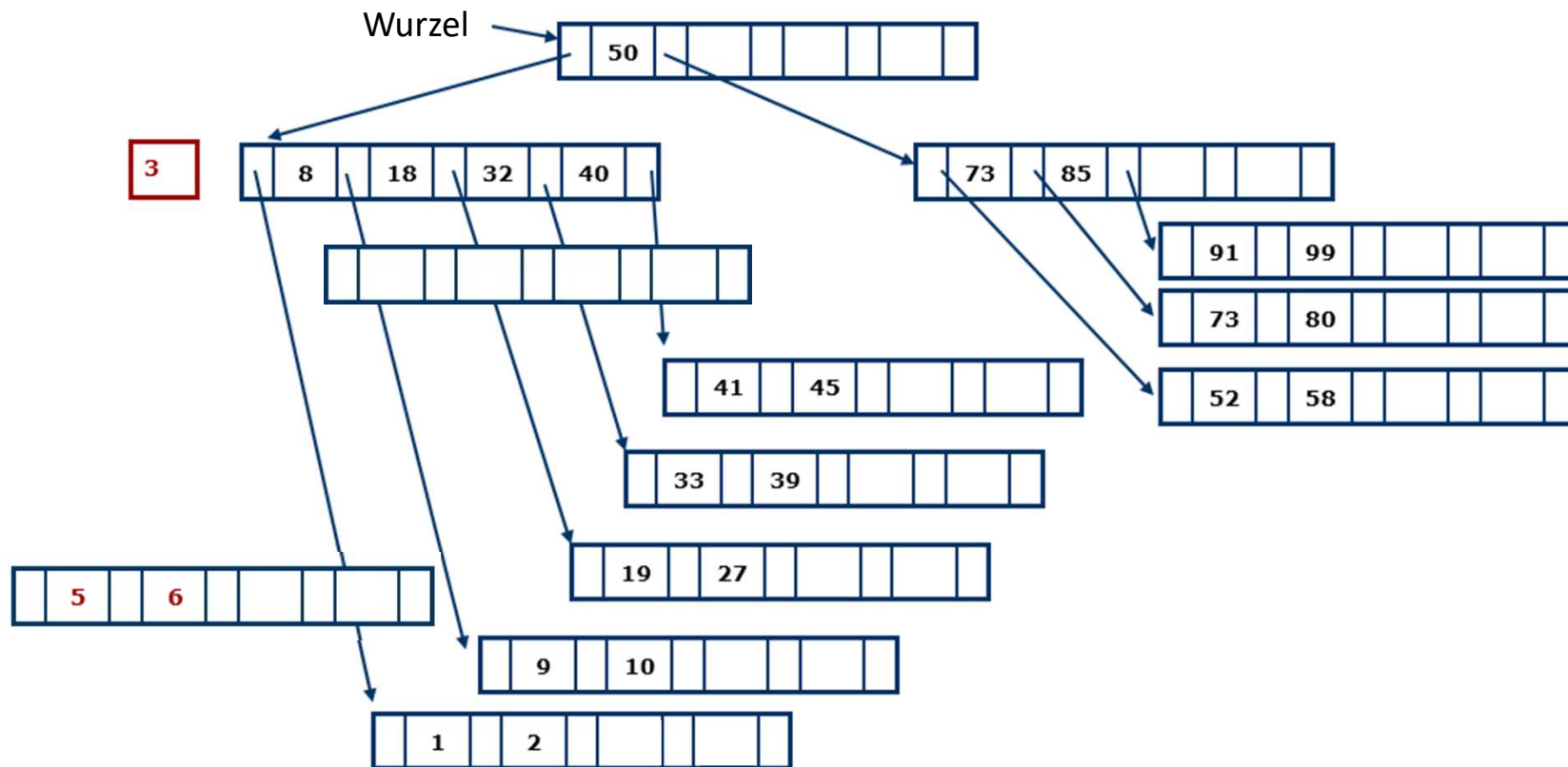


## Aufgabe 9 – füge „3“ ein

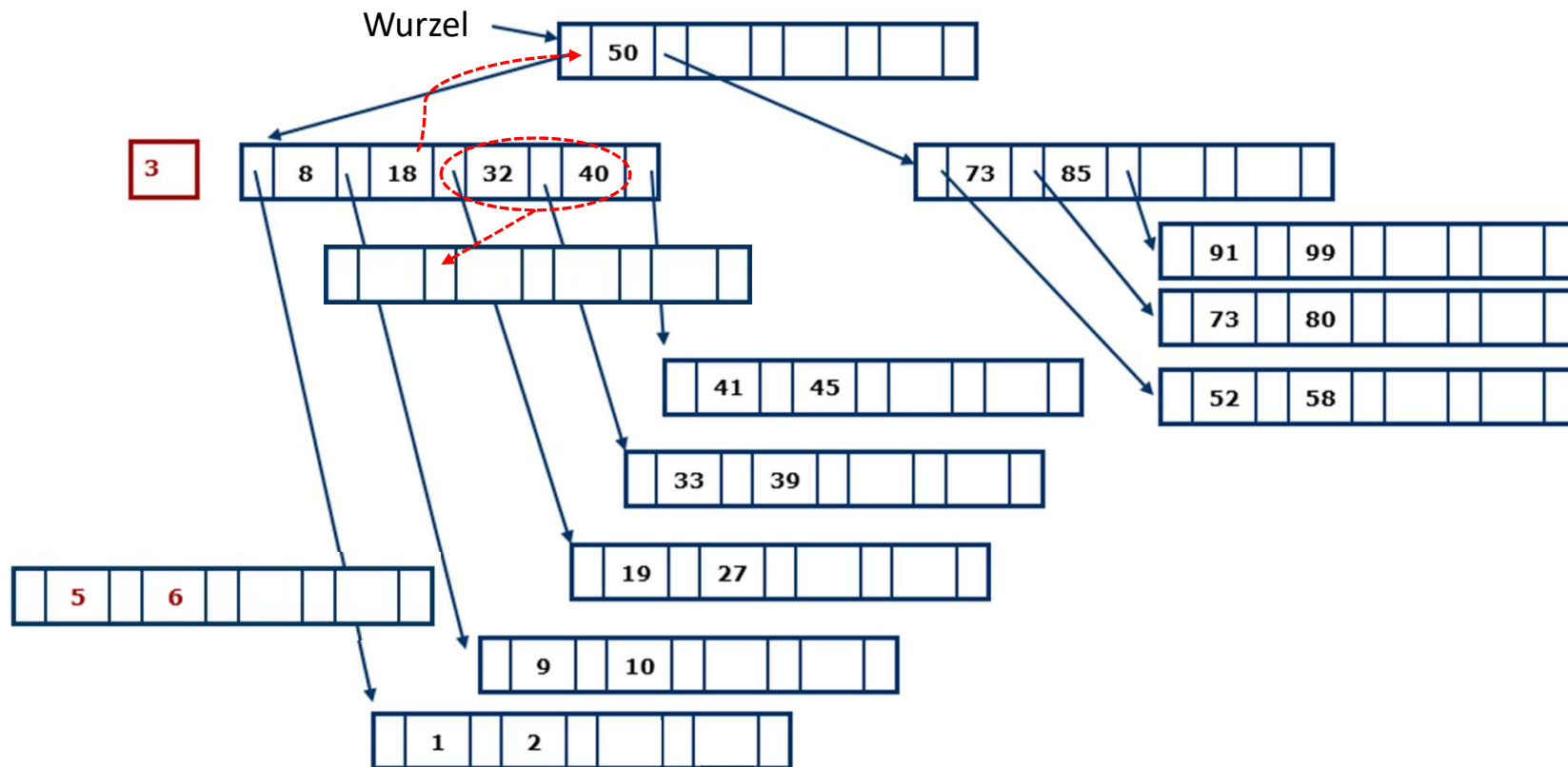




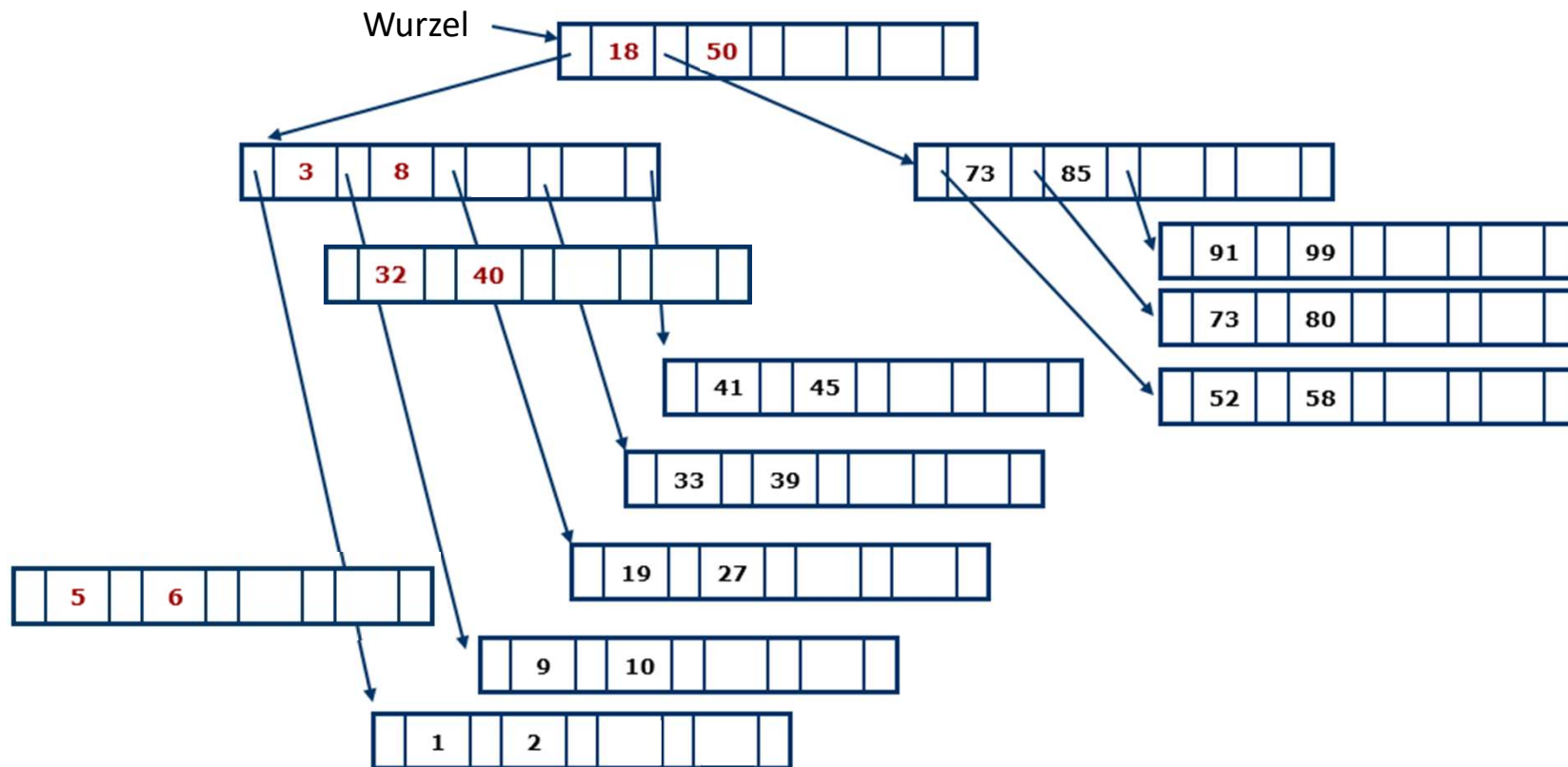
## Aufgabe 9 – füge „3“ ein



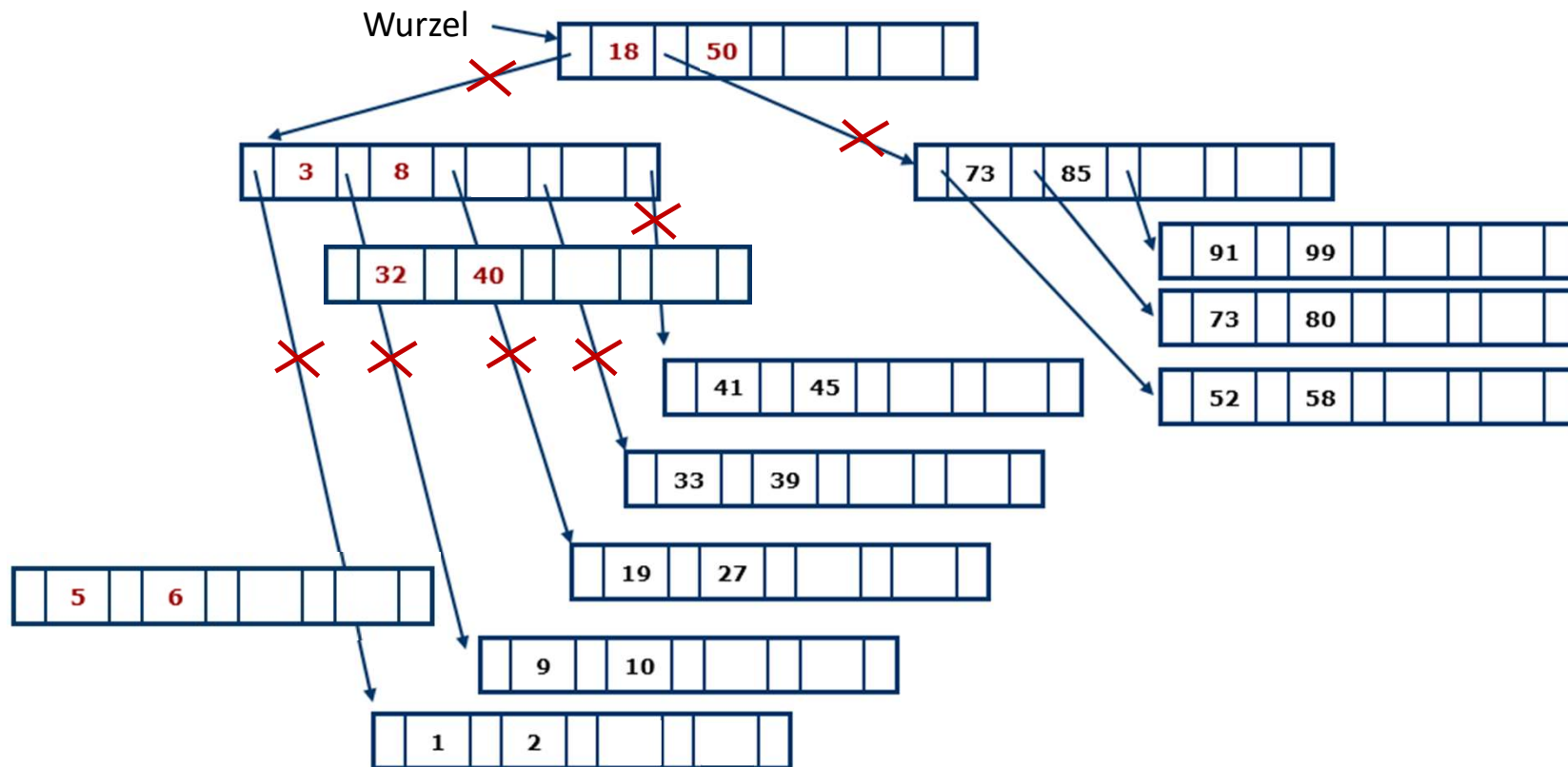
## Aufgabe 9 – füge „3“ ein



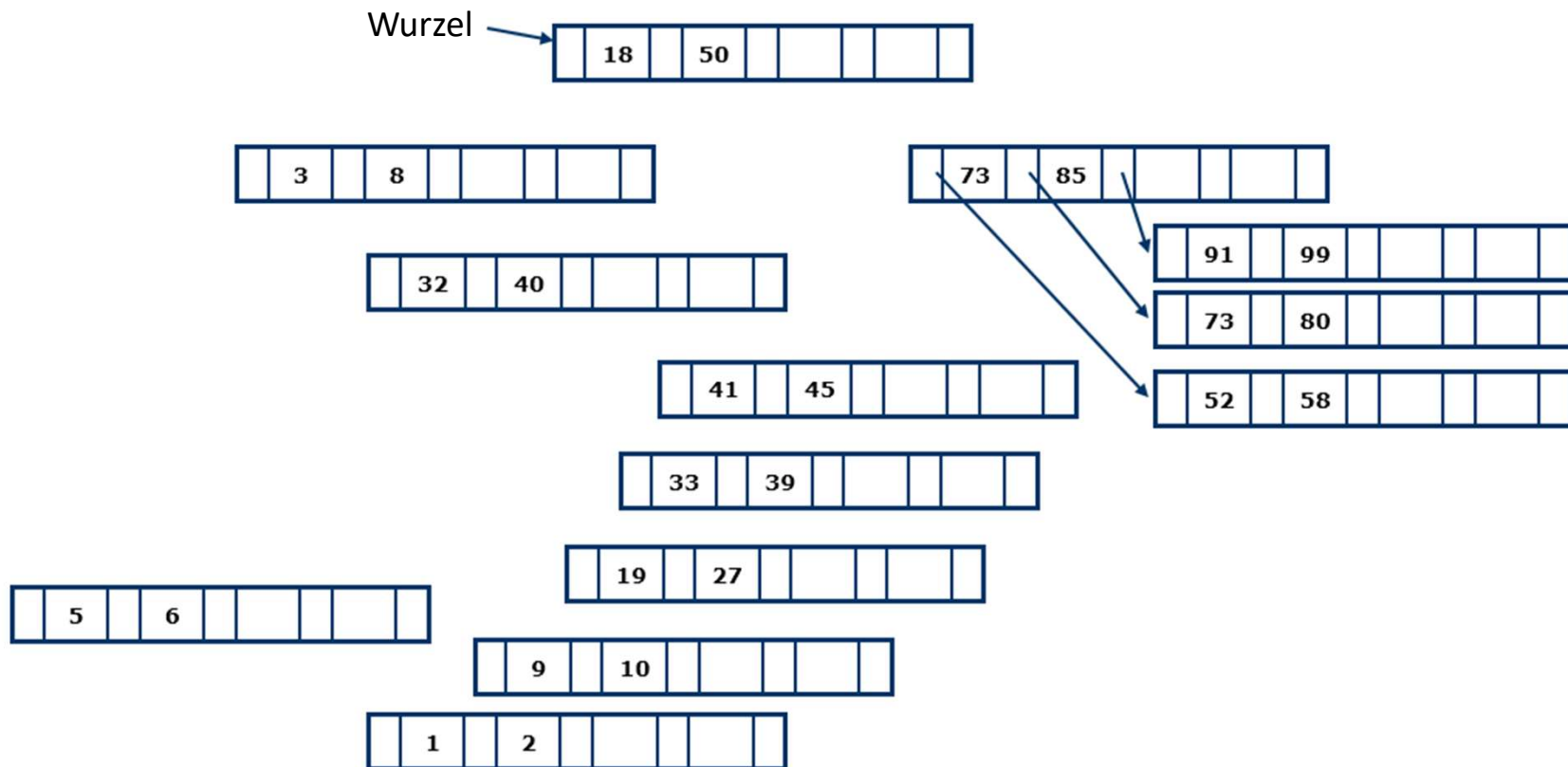
## Aufgabe 9 – füge „3“ ein



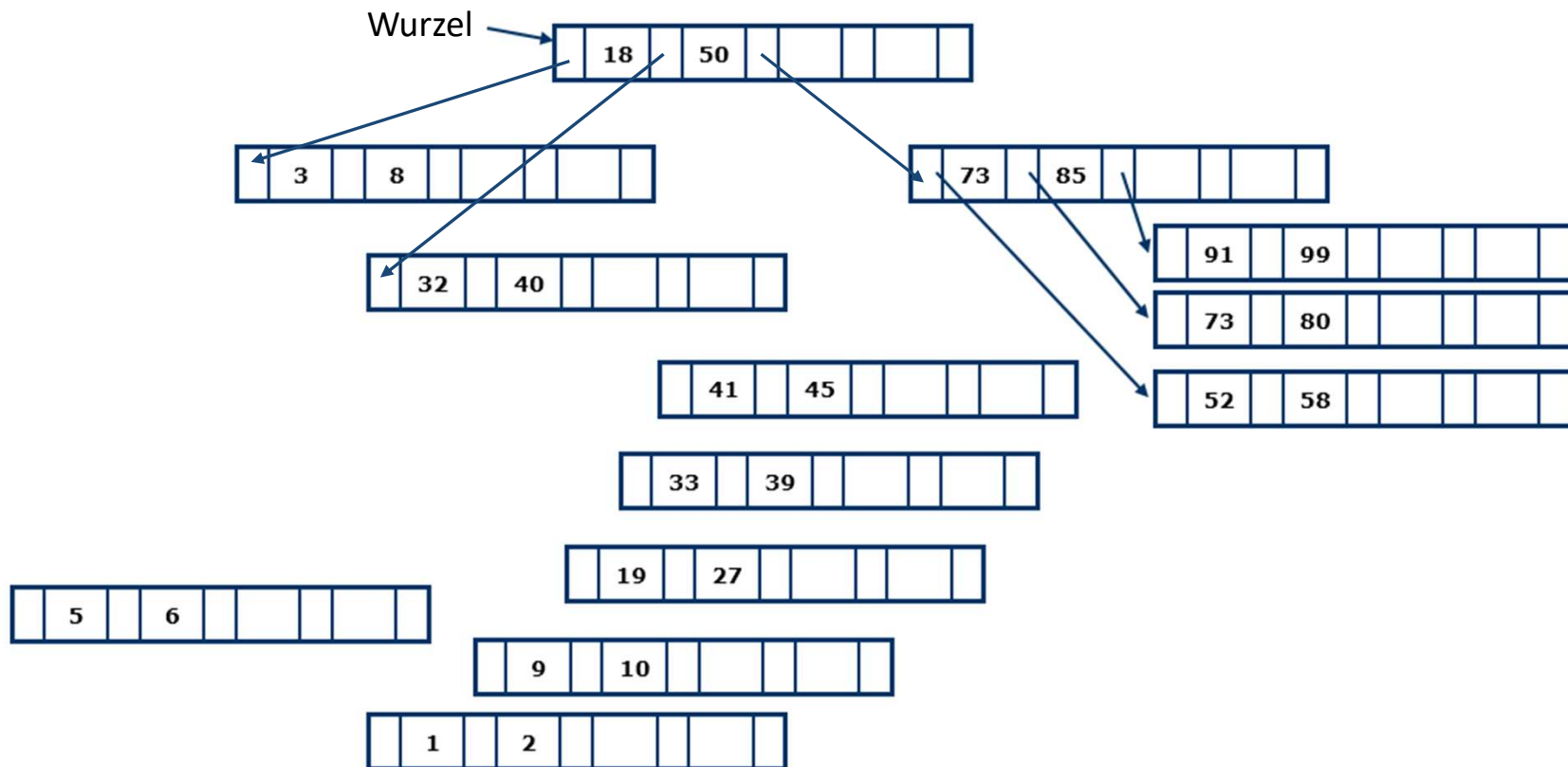
## Aufgabe 9 – füge „3“ ein



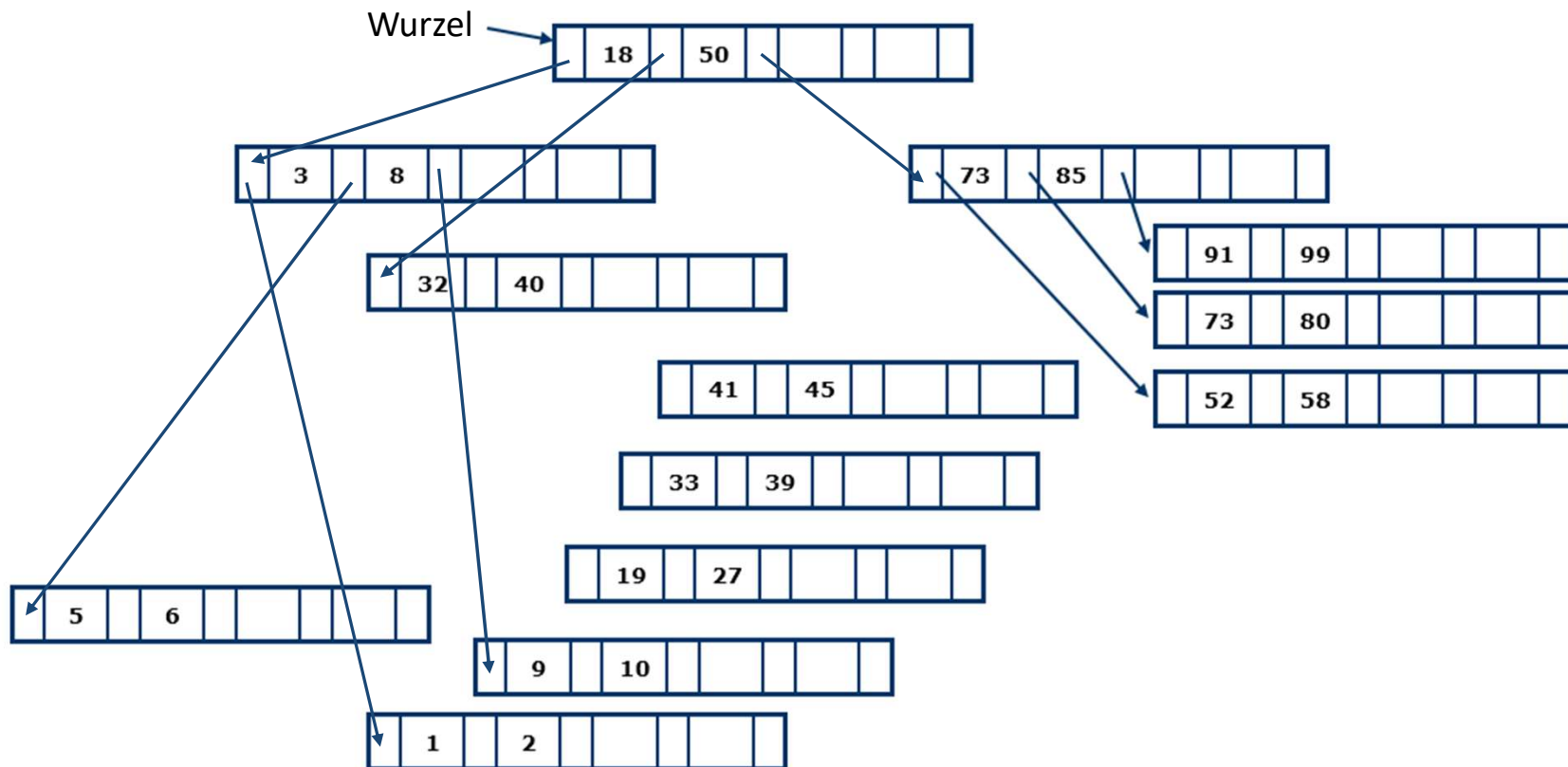
## Aufgabe 9 – füge „3“ ein



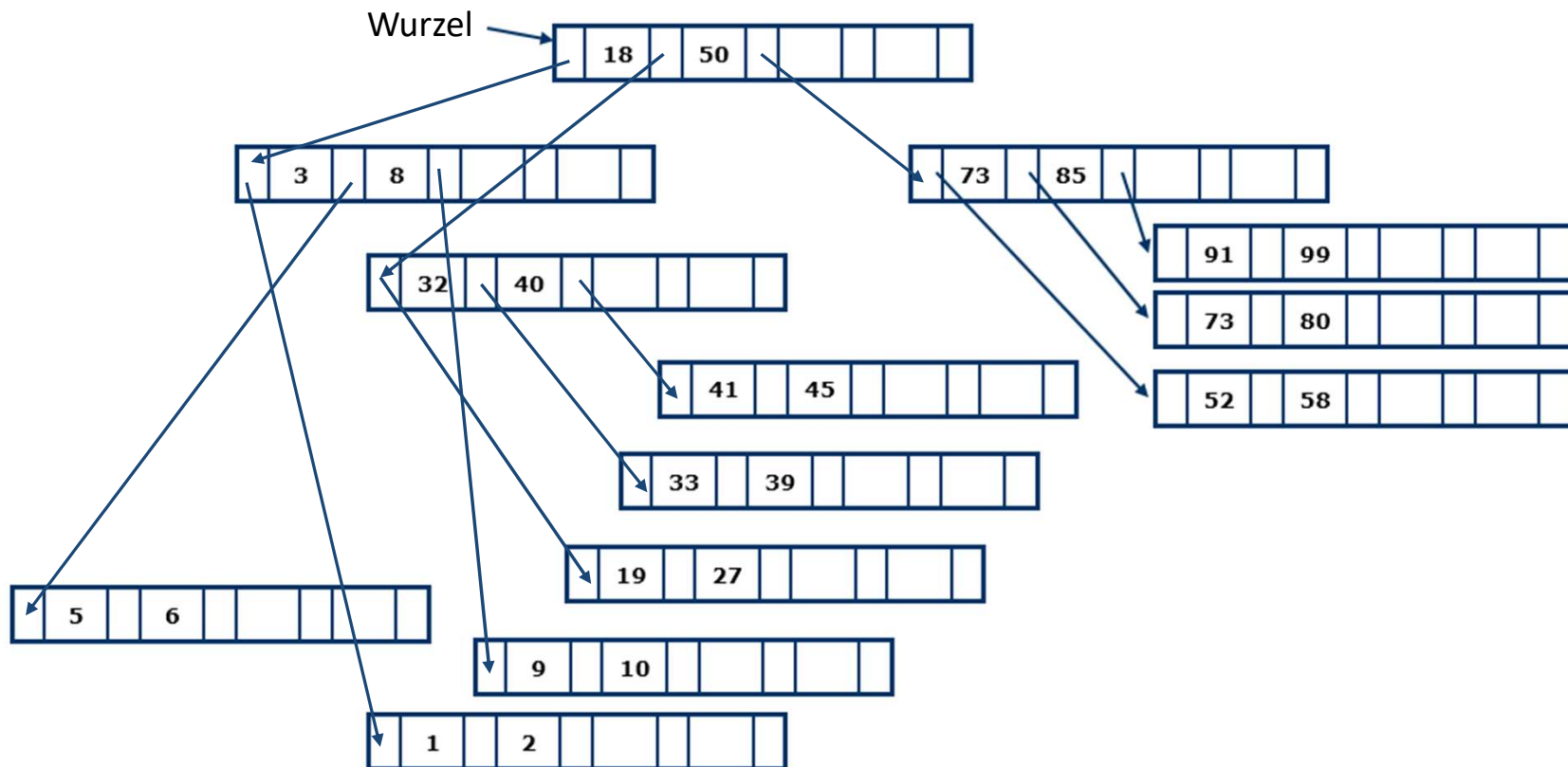
## Aufgabe 9 – füge „3“ ein



## Aufgabe 9 – füge „3“ ein

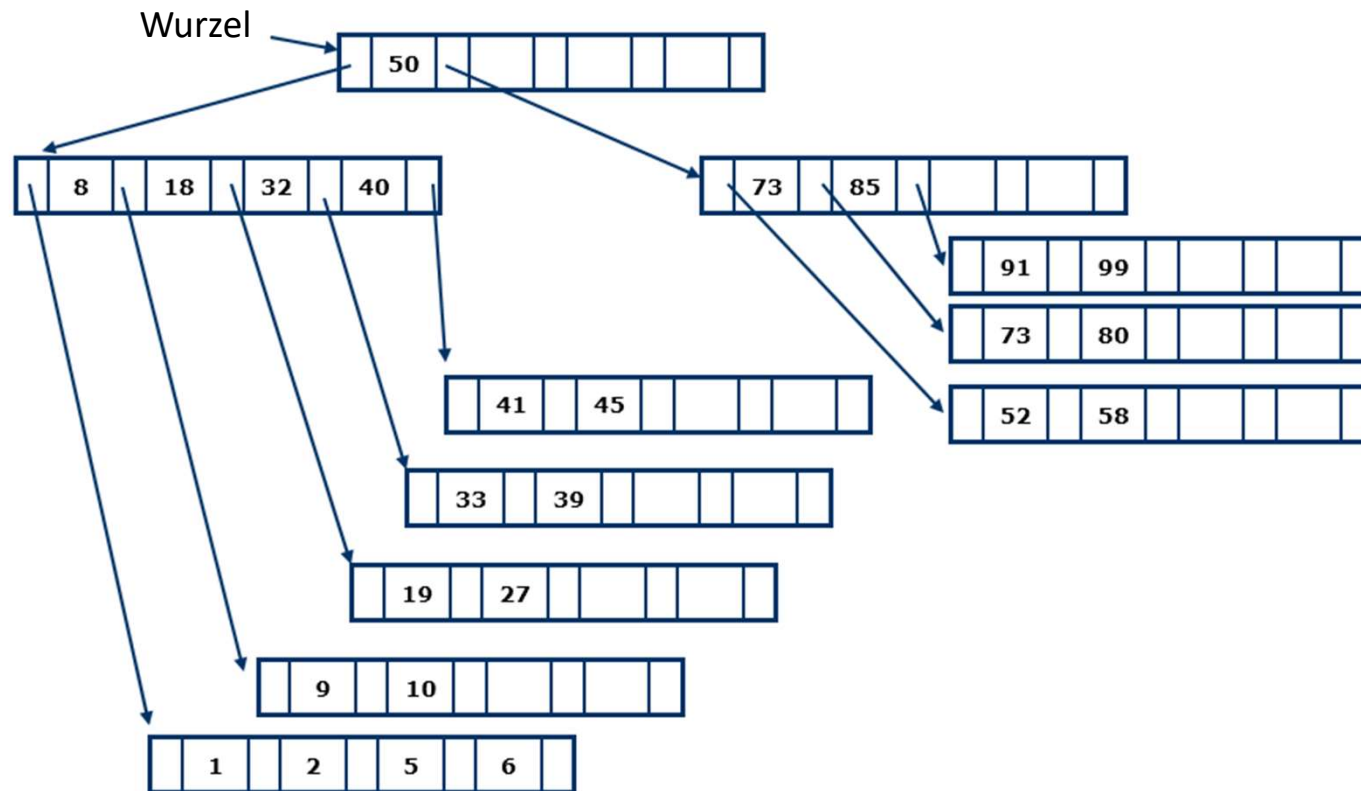


## Aufgabe 9 – füge „3“ ein

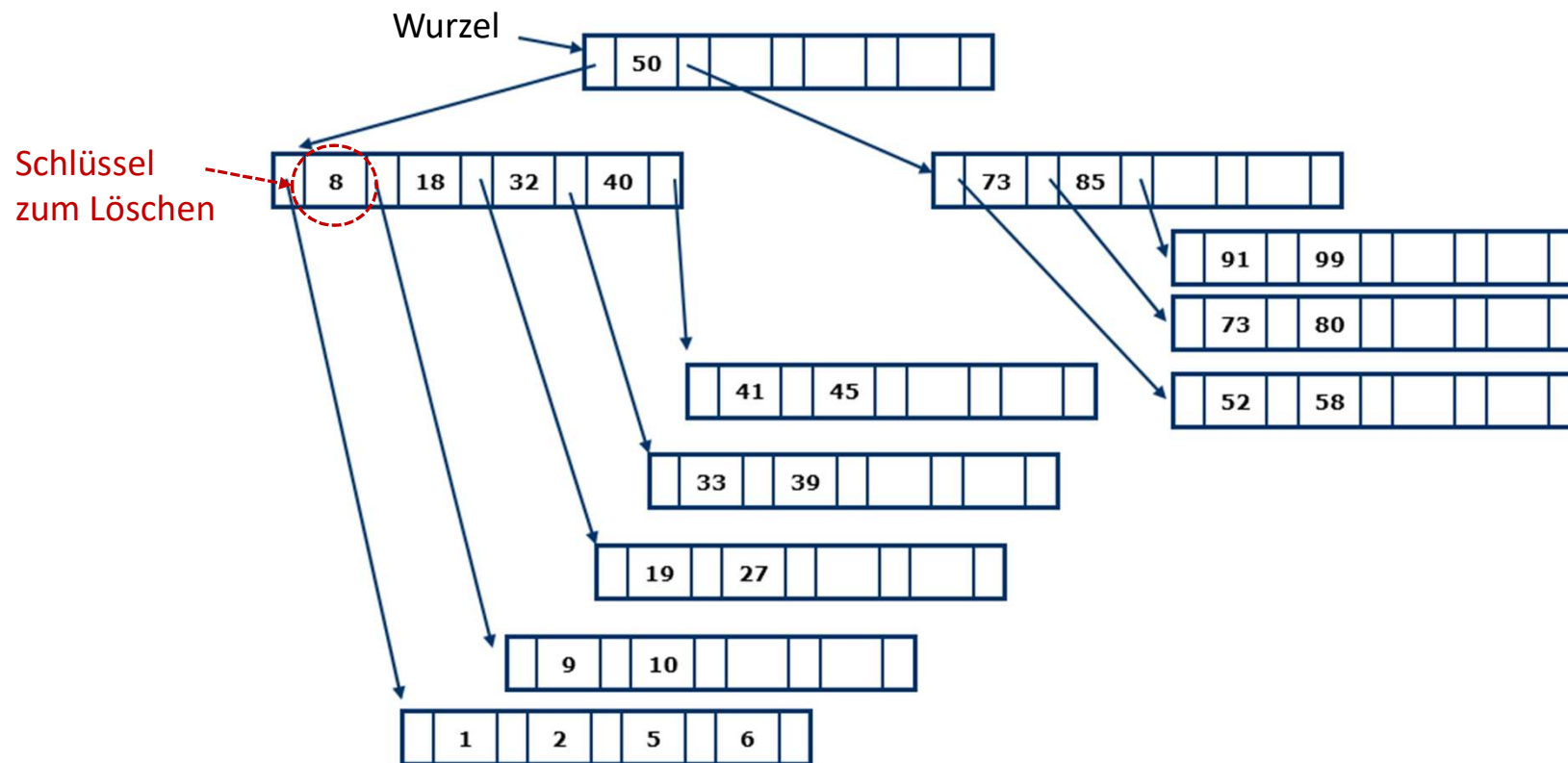




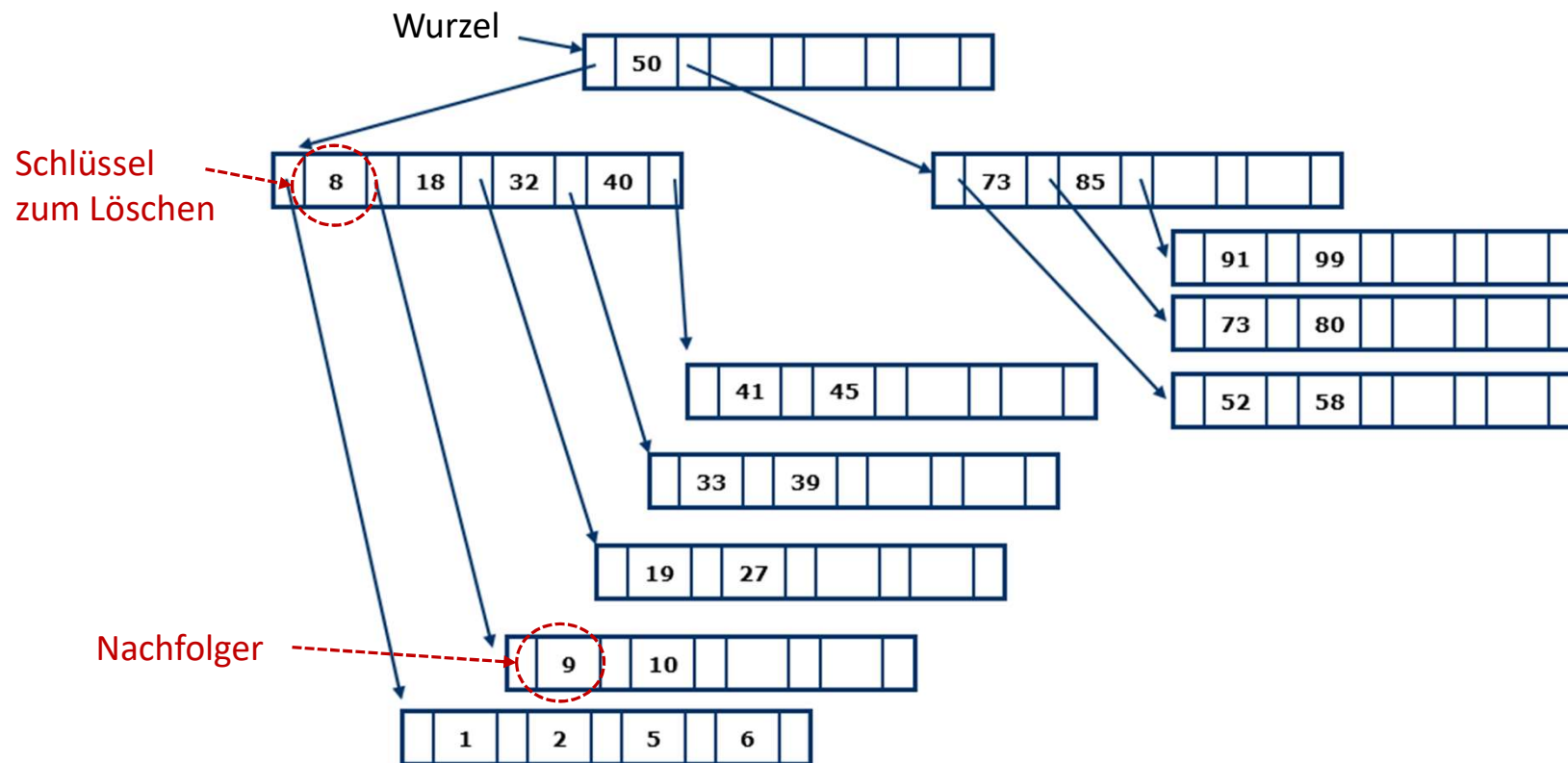
## Aufgabe 9 – lösche „8“ mit Balance-Operationen



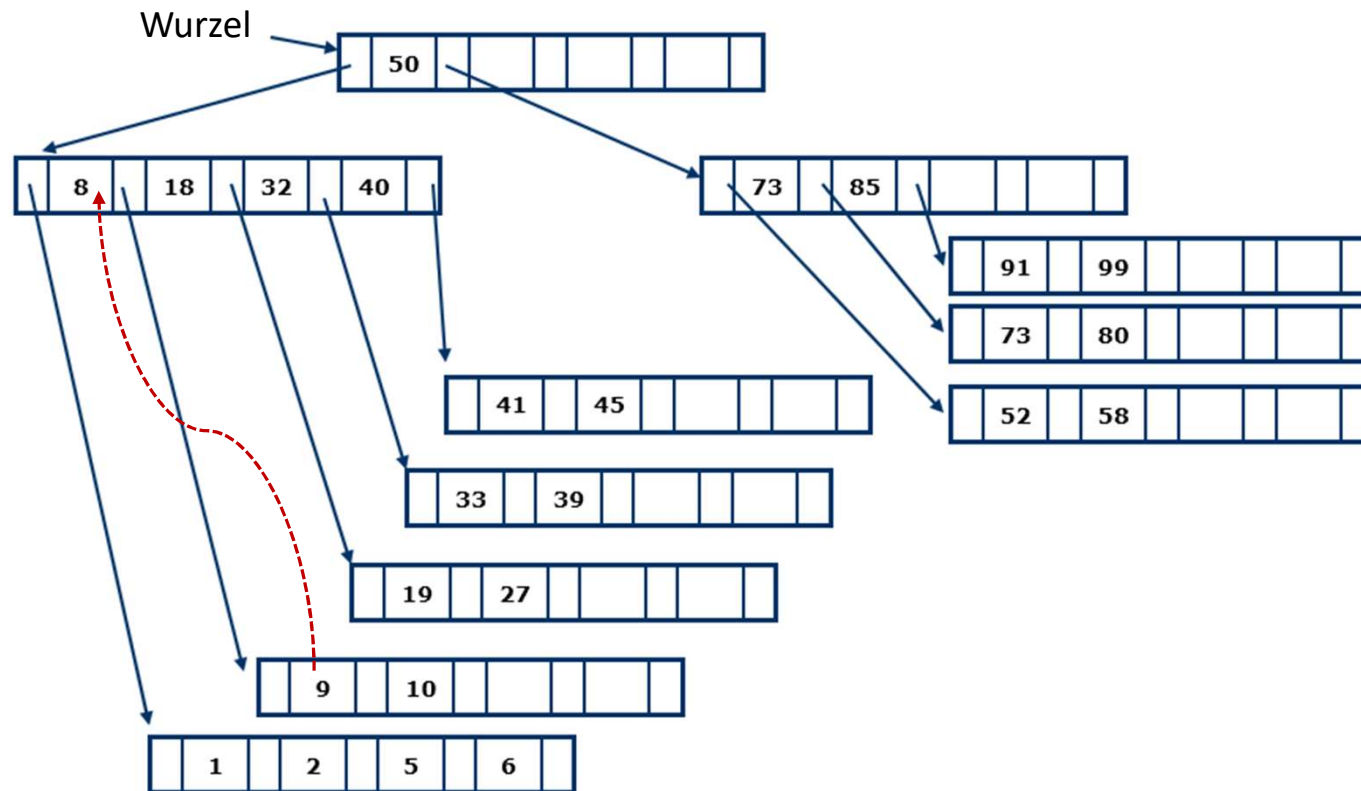
## Aufgabe 9 – lösche „8“ mit Balance-Operationen



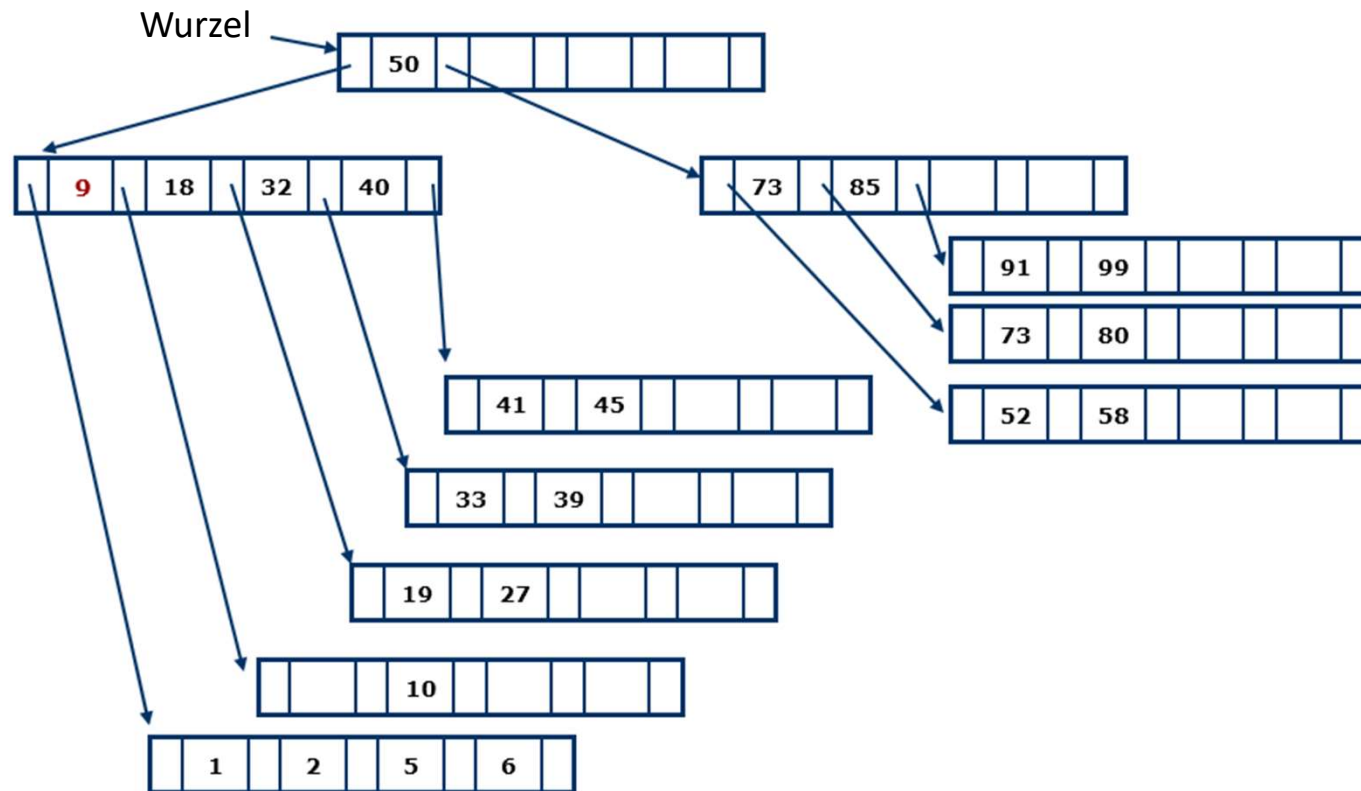
## Aufgabe 9 – lösche „8“ mit Balance-Operationen



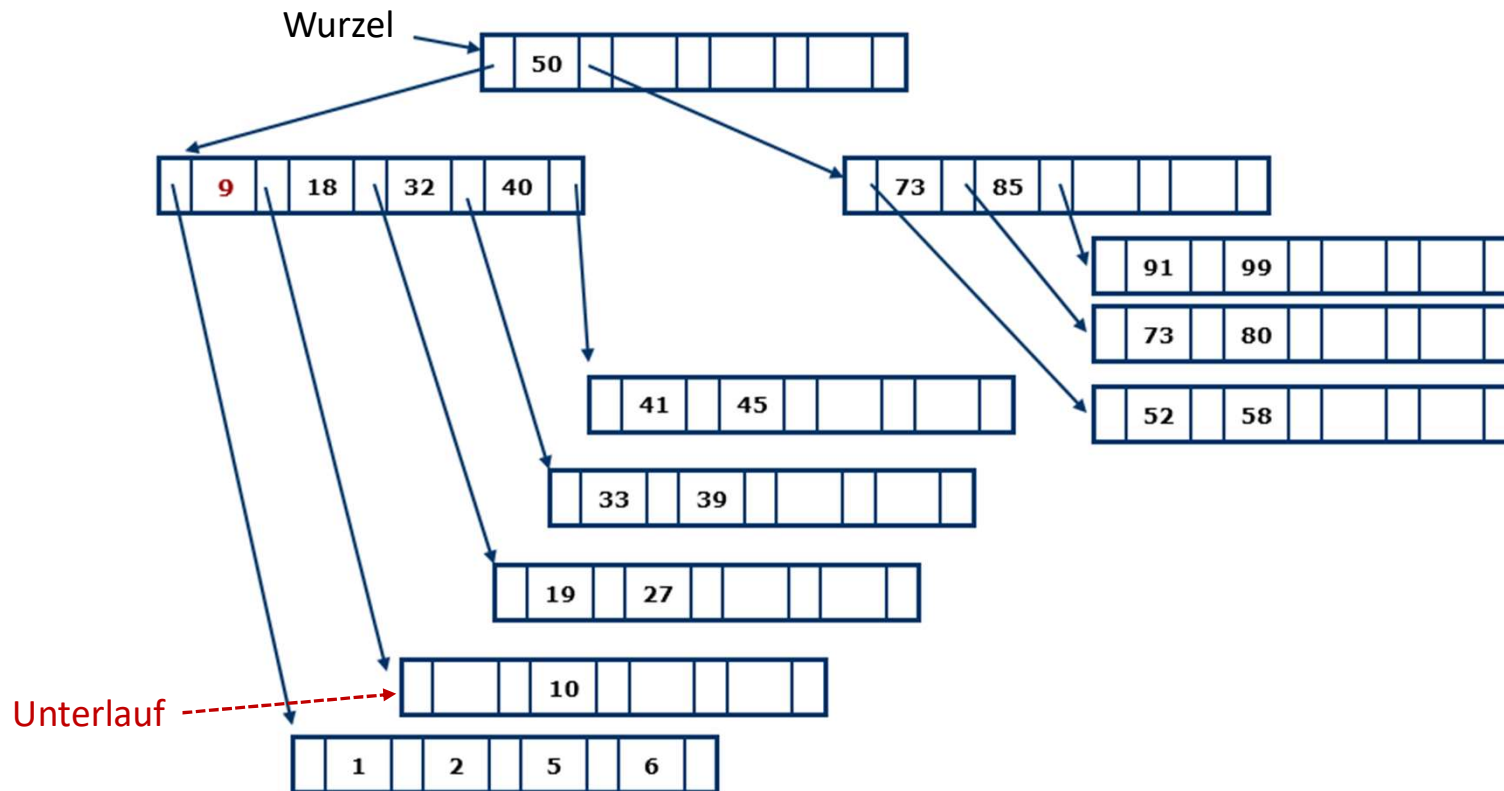
## Aufgabe 9 – lösche „8“ mit Balance-Operationen



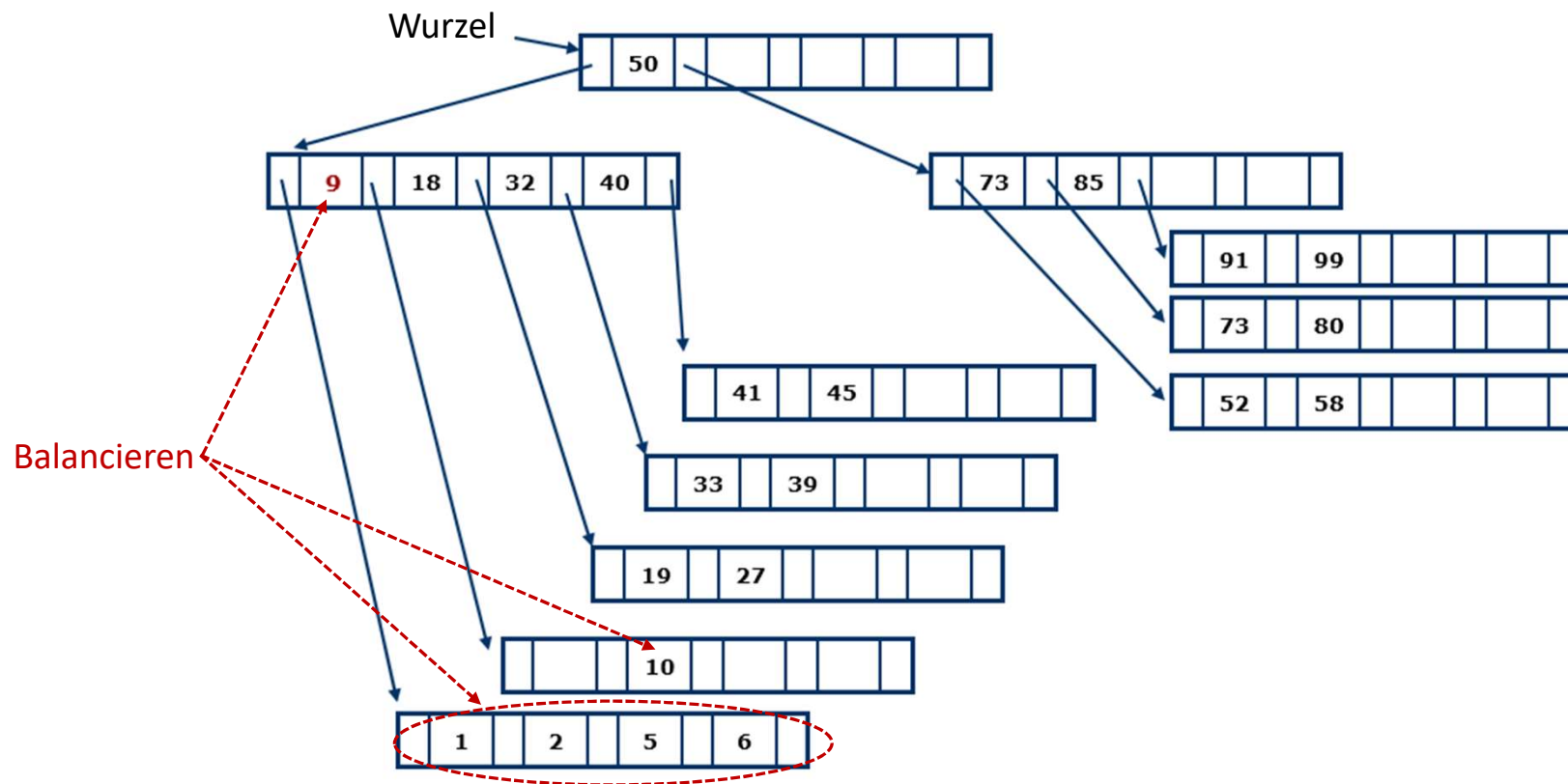
## Aufgabe 9 – lösche „8“ mit Balance-Operationen



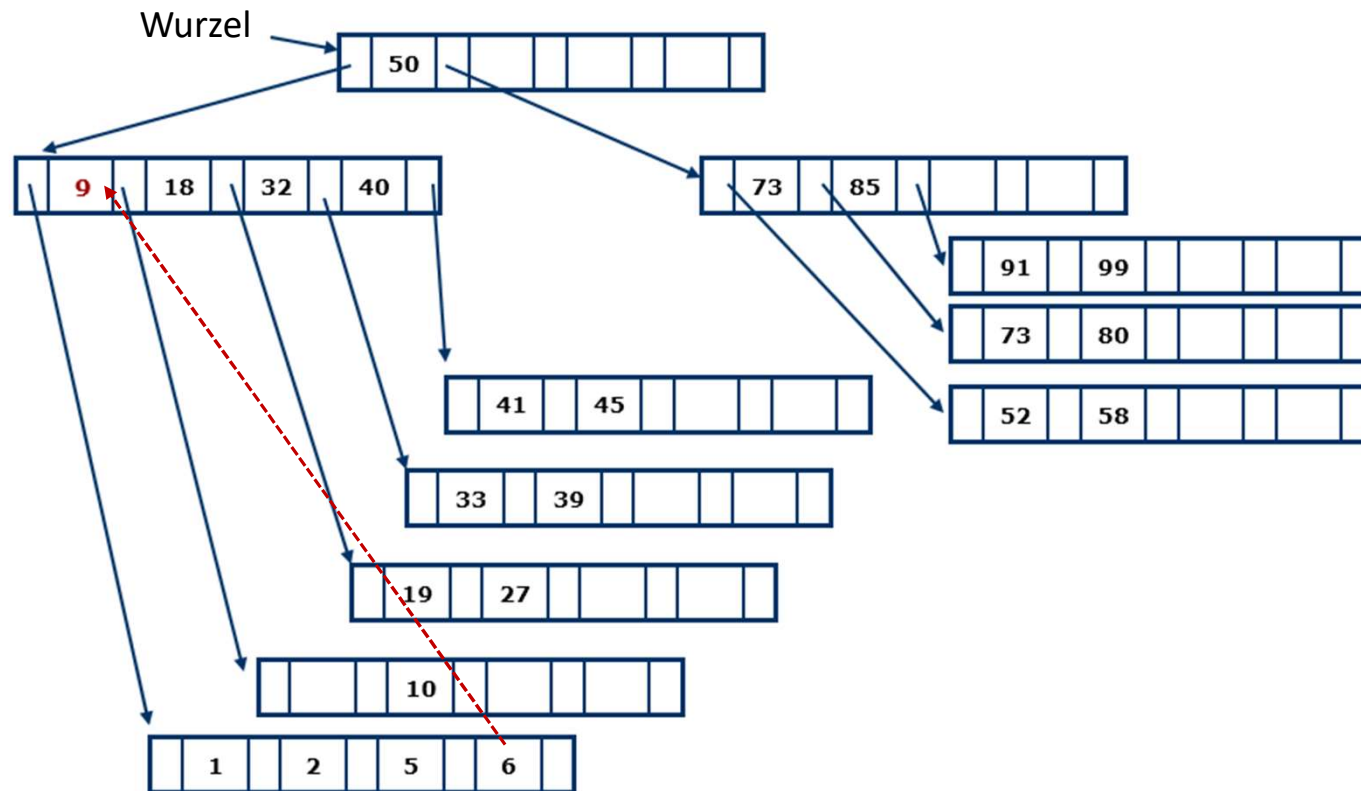
## Aufgabe 9 – lösche „8“ mit Balance-Operationen



## Aufgabe 9 – lösche „8“ mit Balance-Operationen

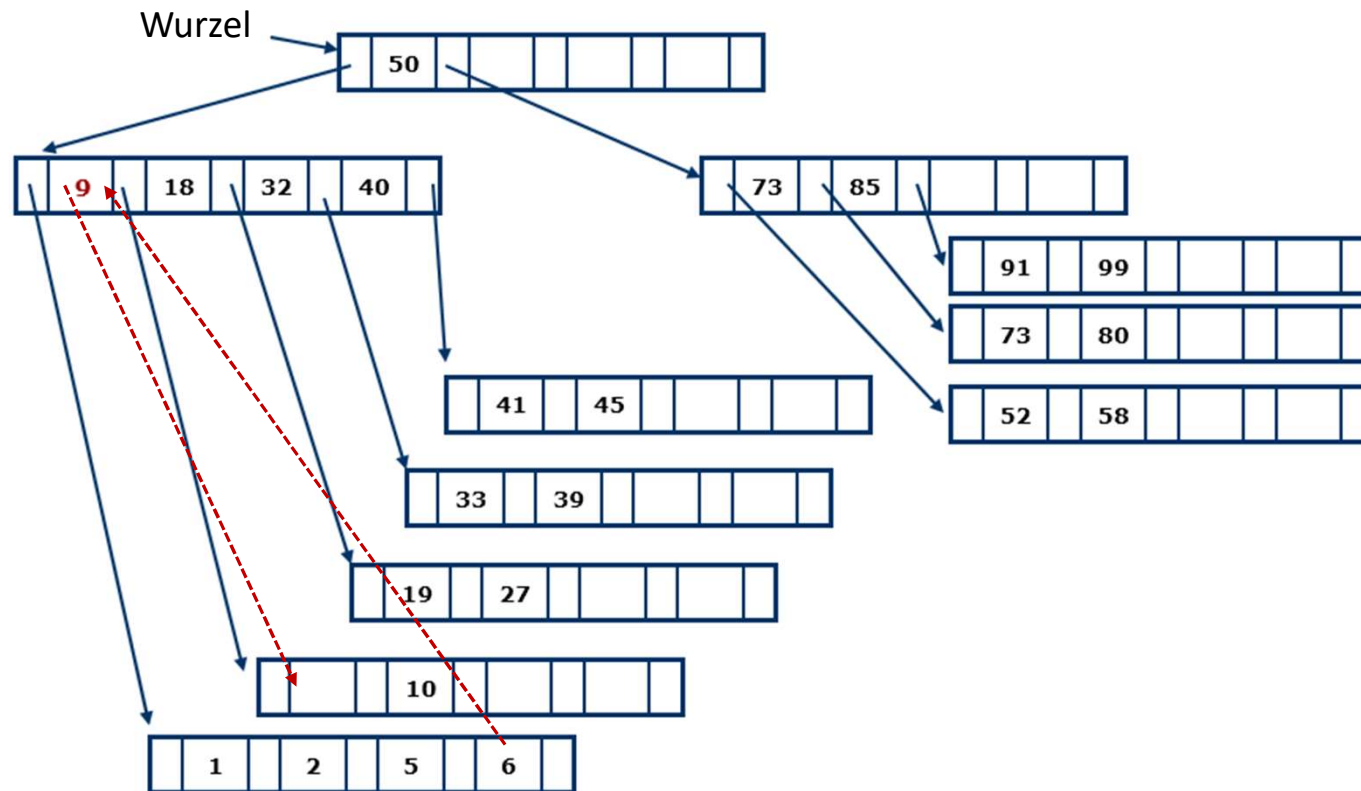


## Aufgabe 9 – lösche „8“ mit Balance-Operationen

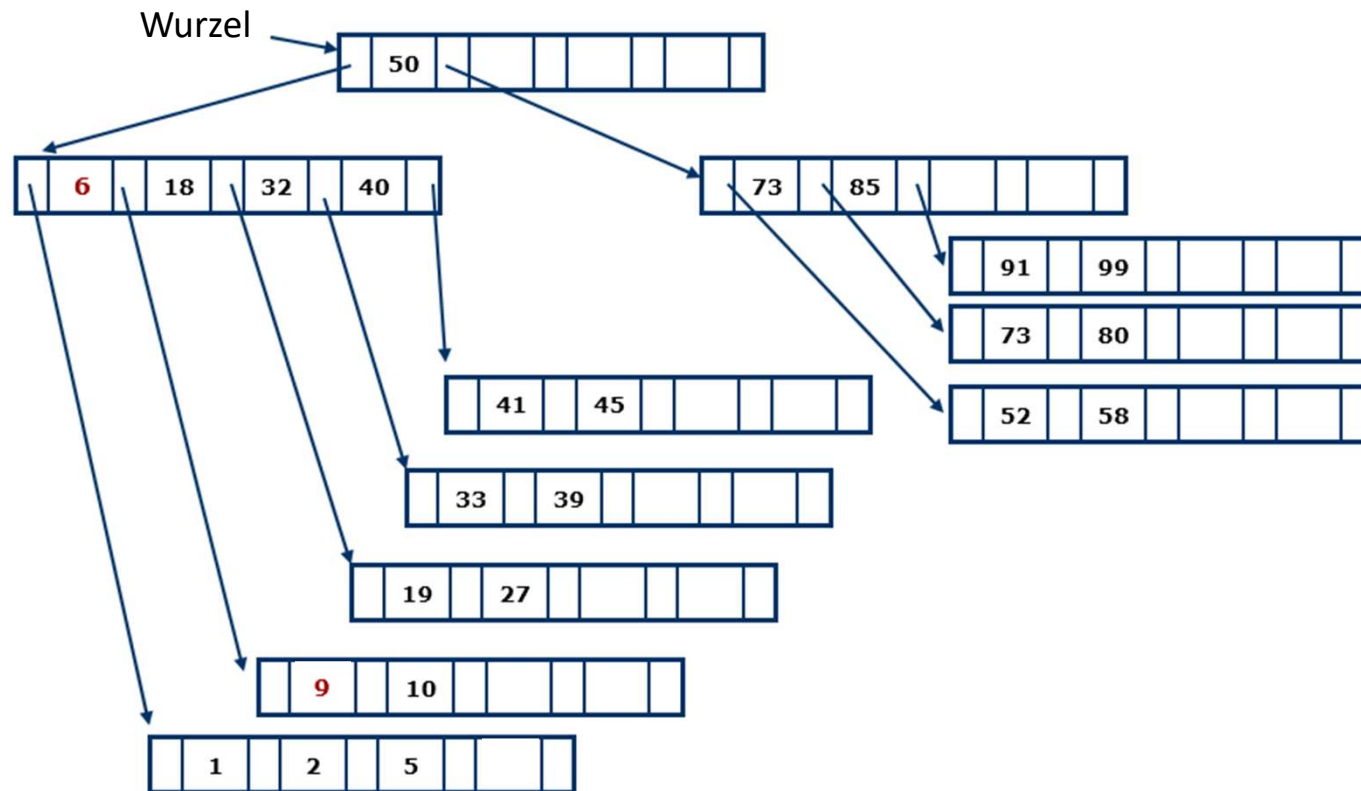




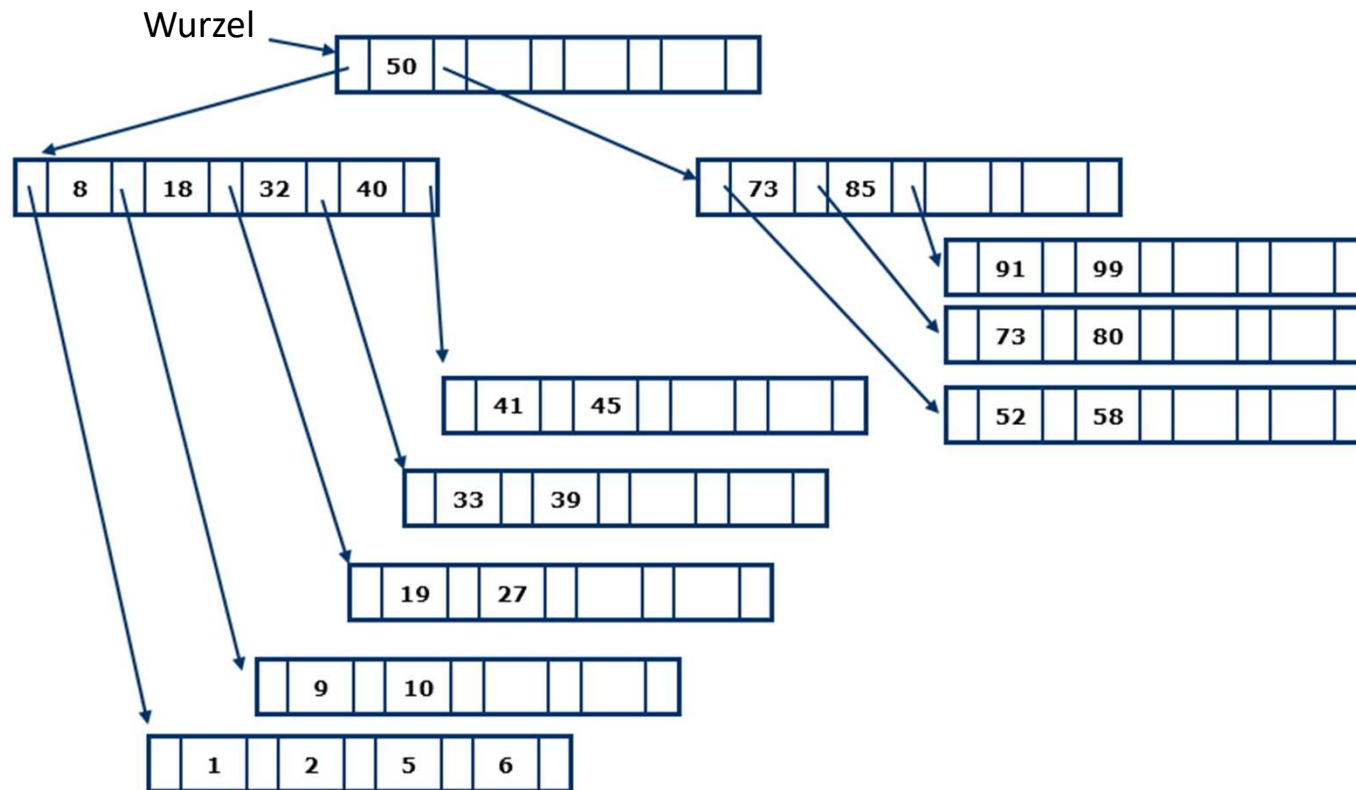
## Aufgabe 9 – lösche „8“ mit Balance-Operationen



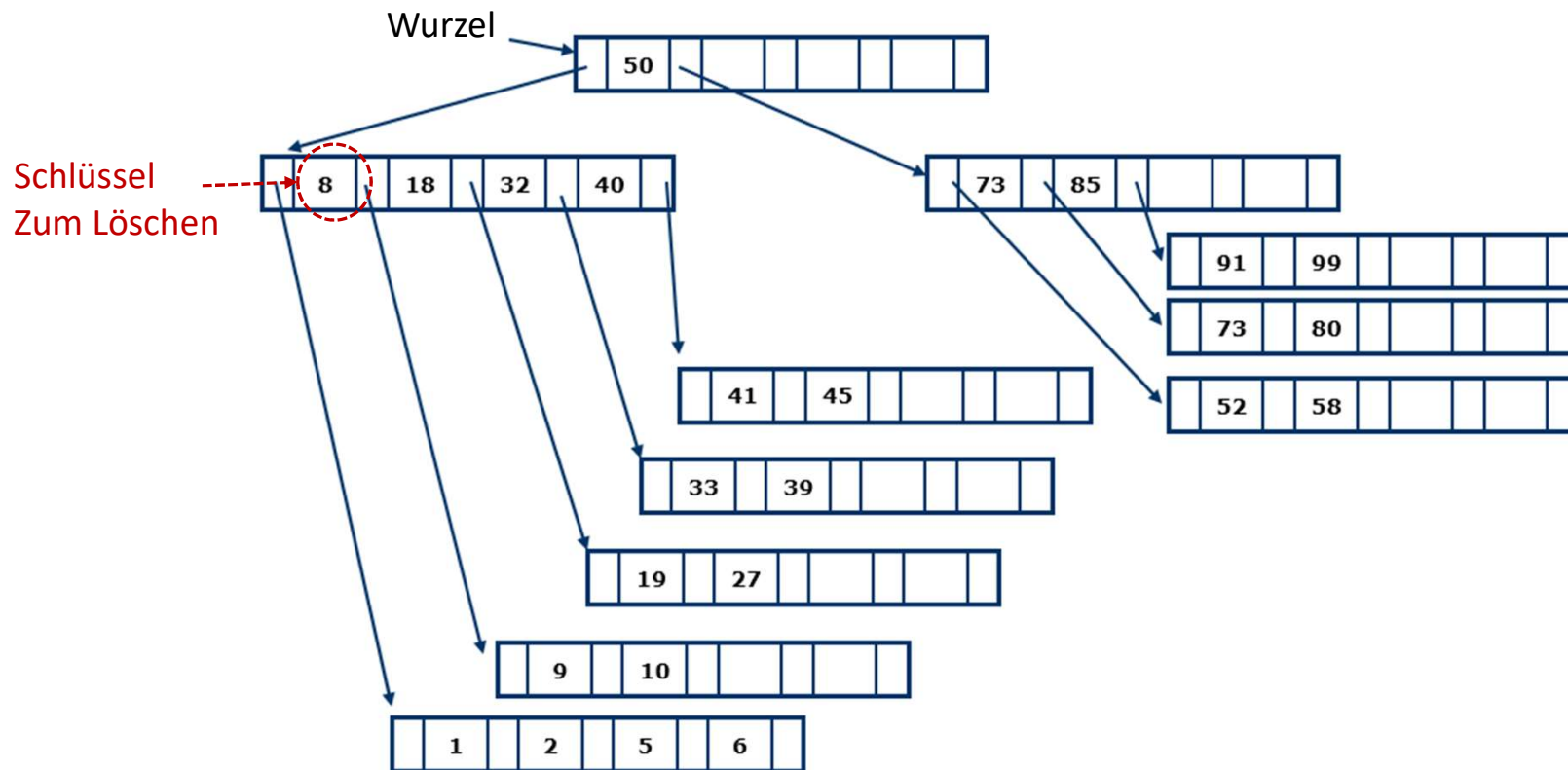
## Aufgabe 9 – lösche „8“ mit Balance-Operationen



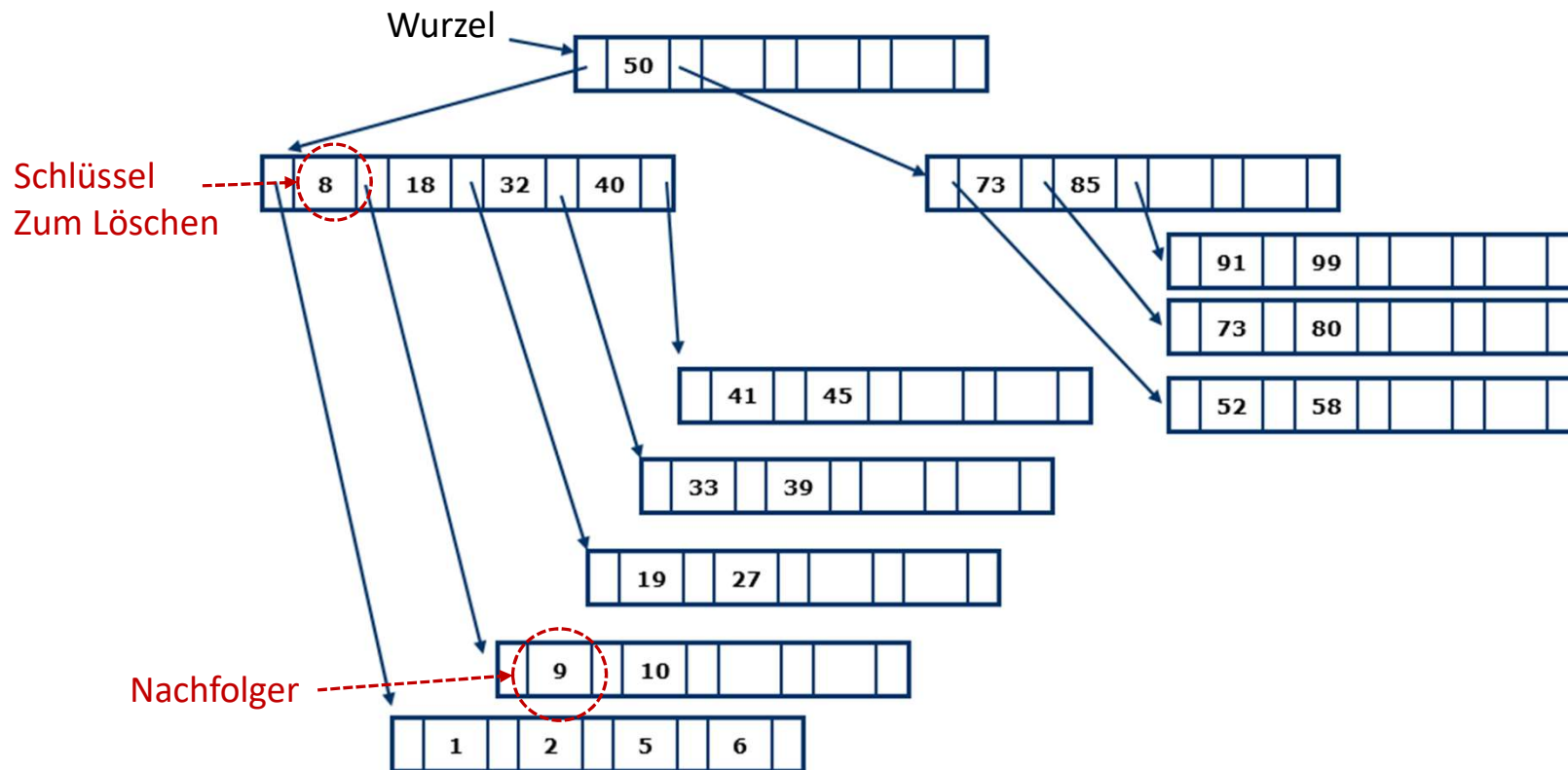
## Aufgabe 9 – lösche „8“ mit Merge-Operationen



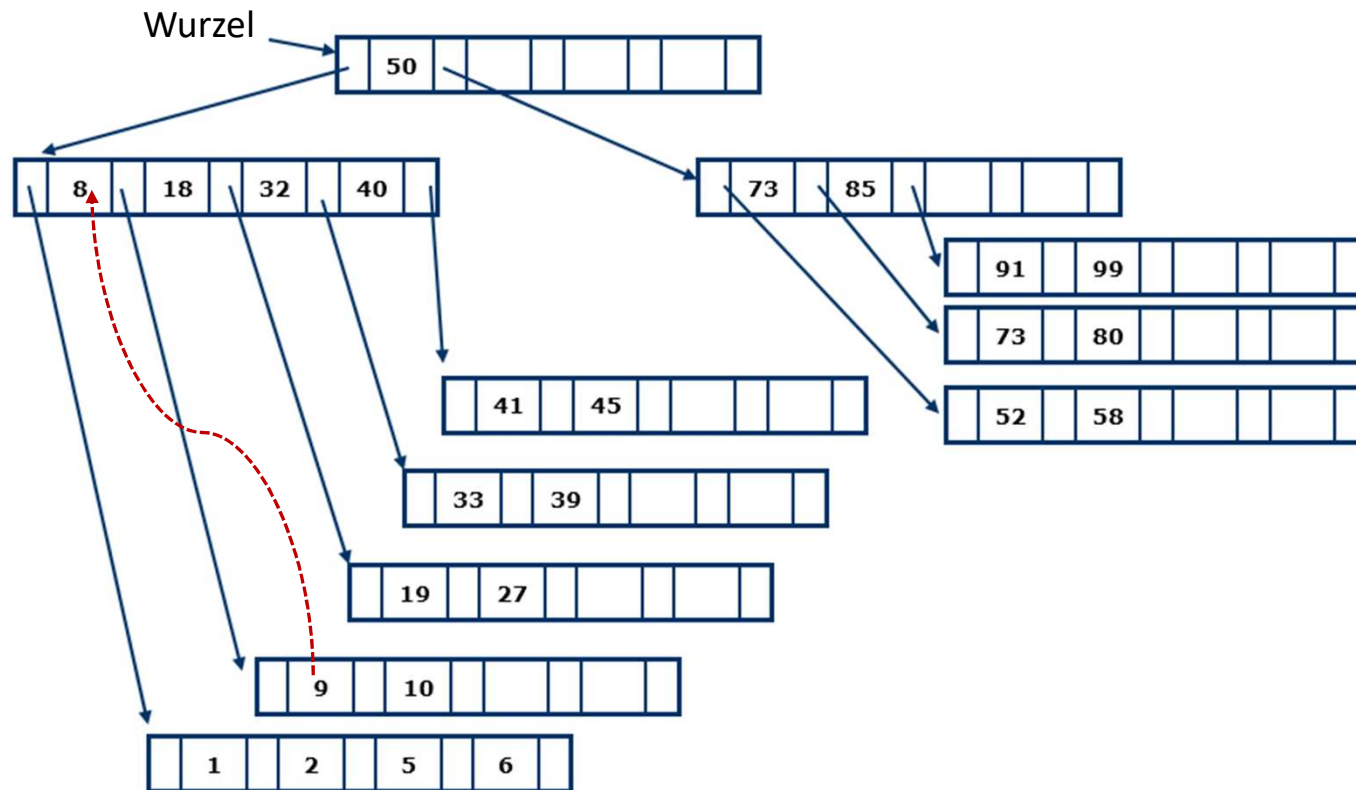
## Aufgabe 9 – lösche „8“ mit Merge-Operationen



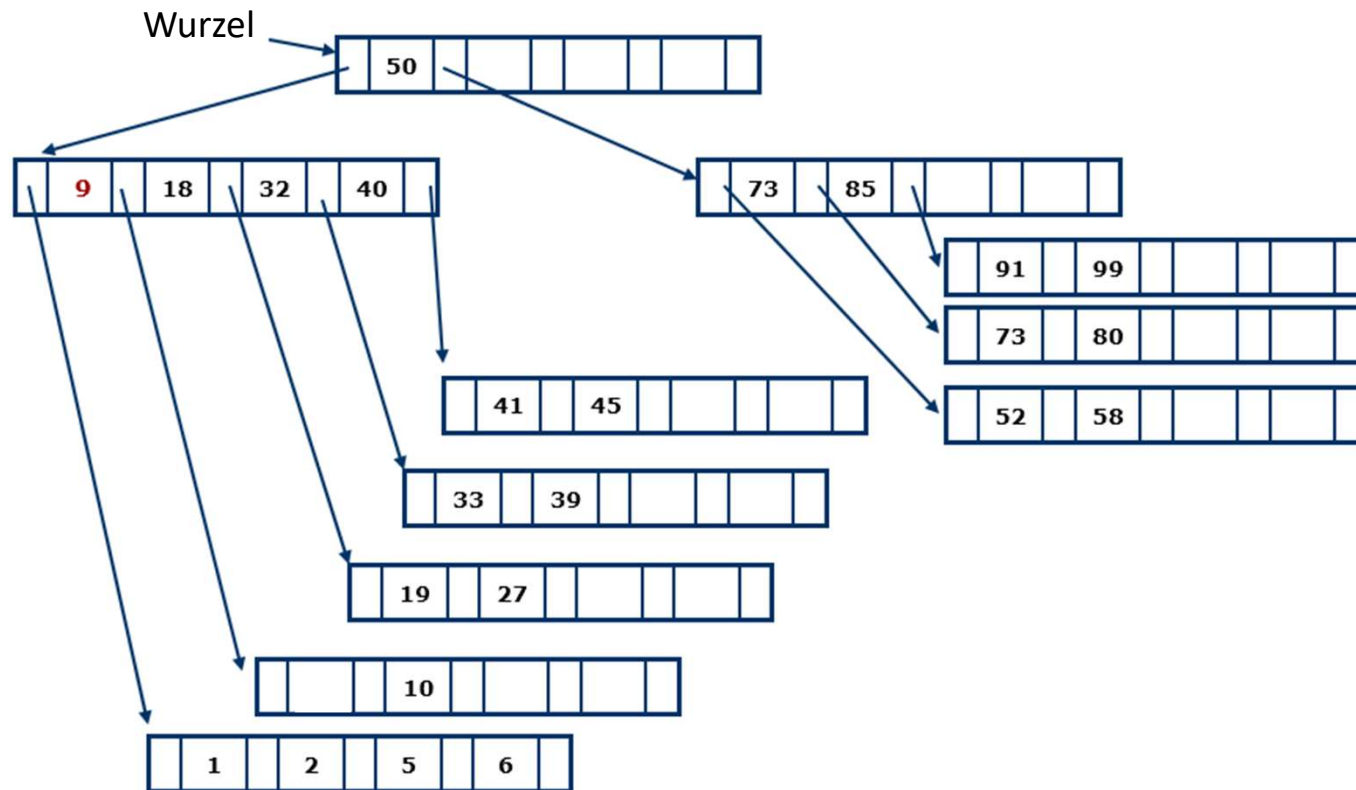
# Aufgabe 9 – lösche „8“ mit Merge-Operationen



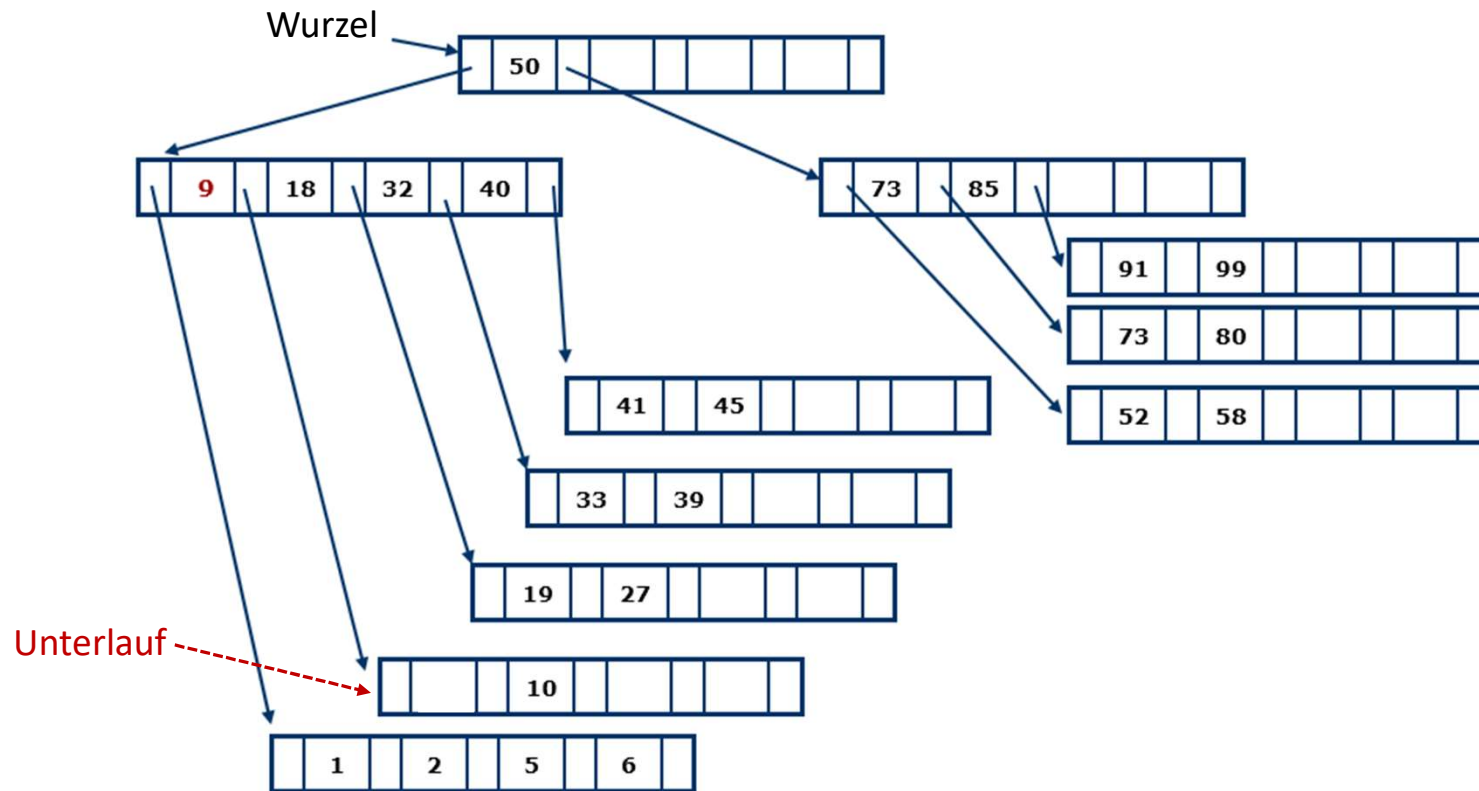
## Aufgabe 9 – lösche „8“ mit Merge-Operationen



## Aufgabe 9 – lösche „8“ mit Merge-Operationen

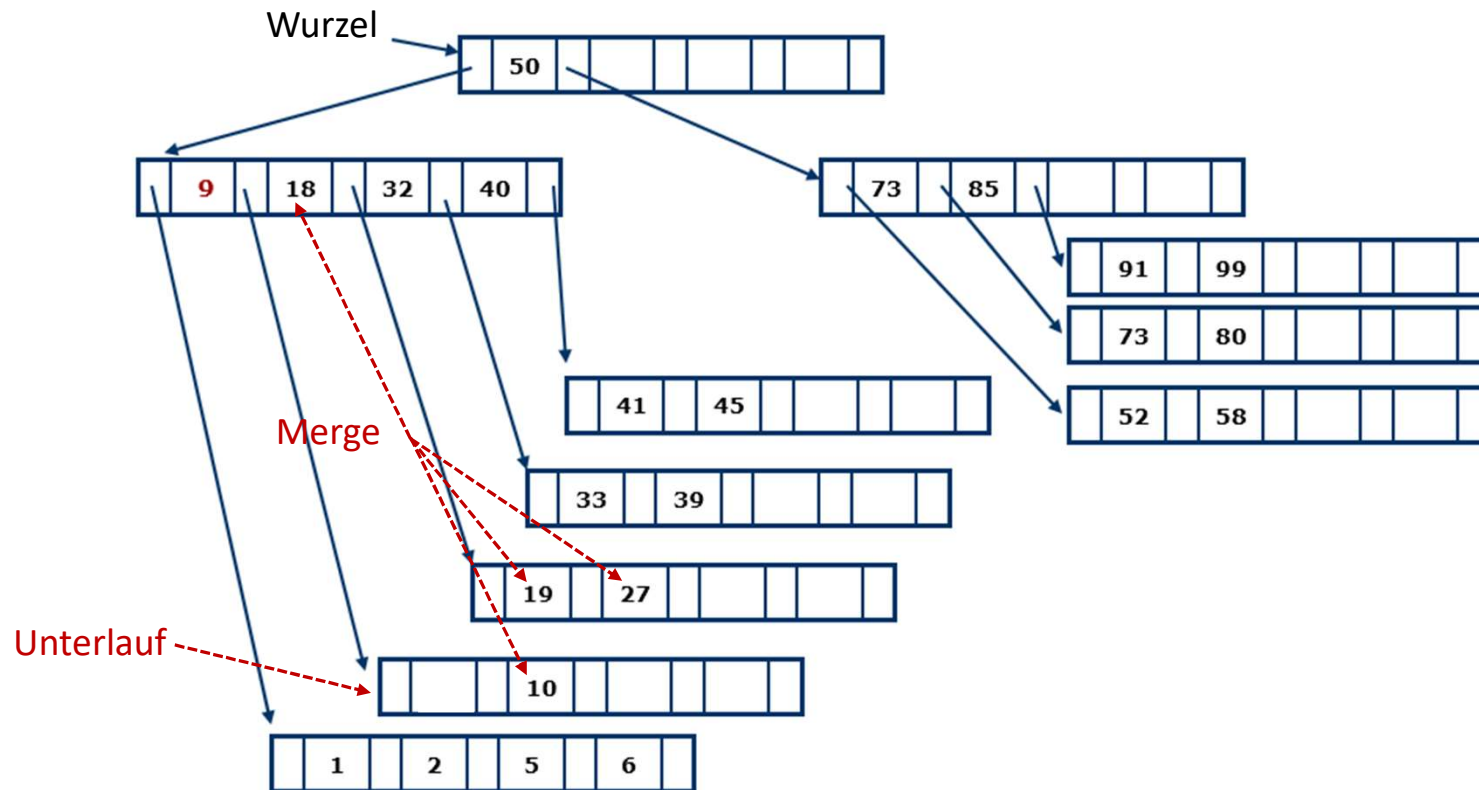


## Aufgabe 9 – lösche „8“ mit Merge-Operationen

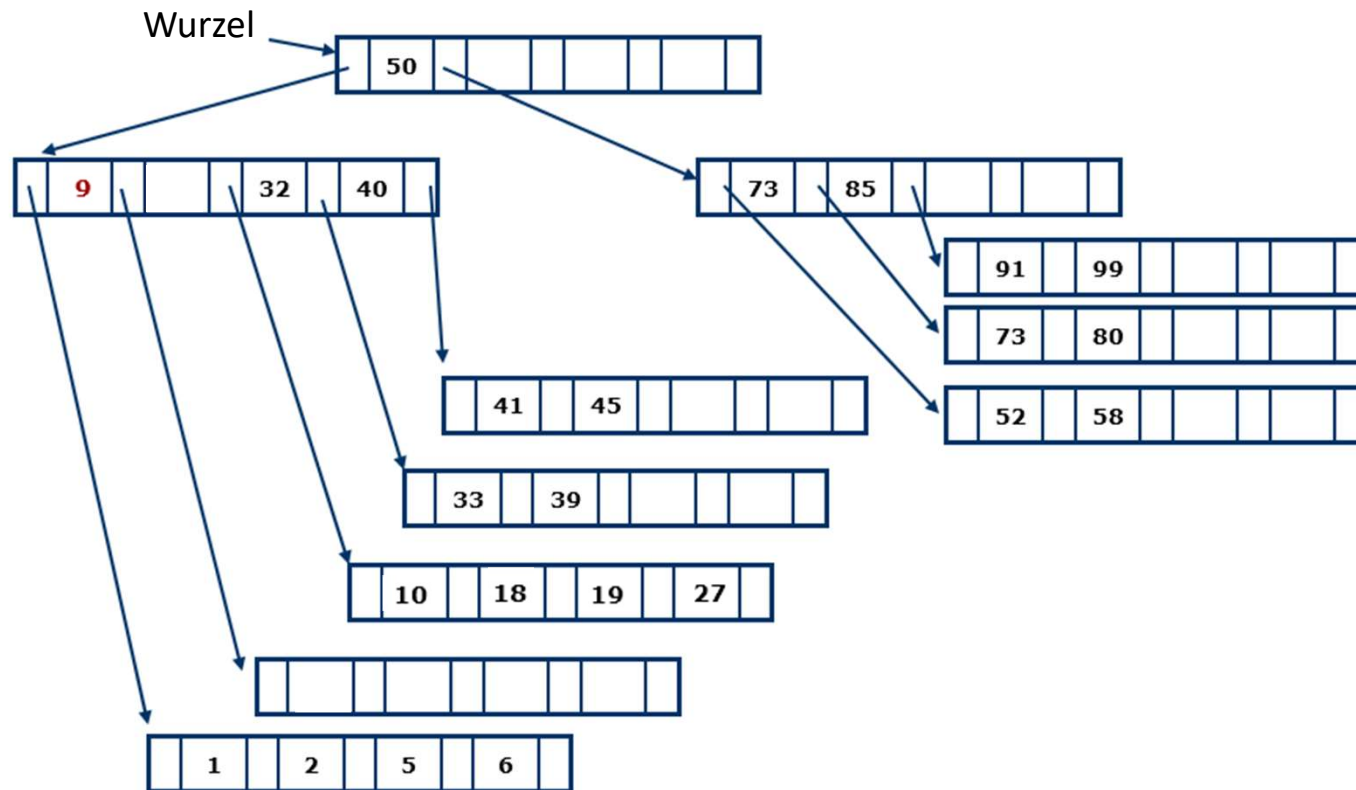




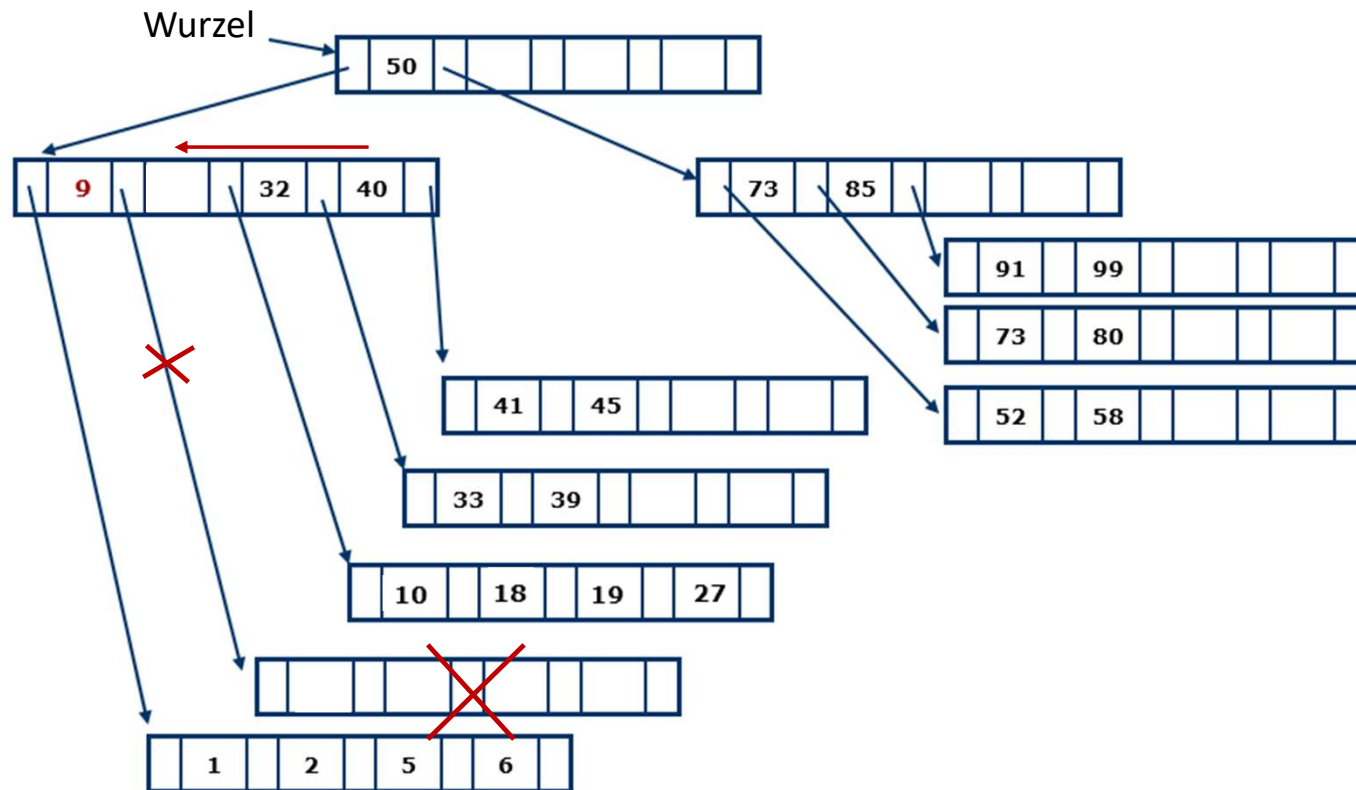
## Aufgabe 9 – lösche „8“ mit Merge-Operationen



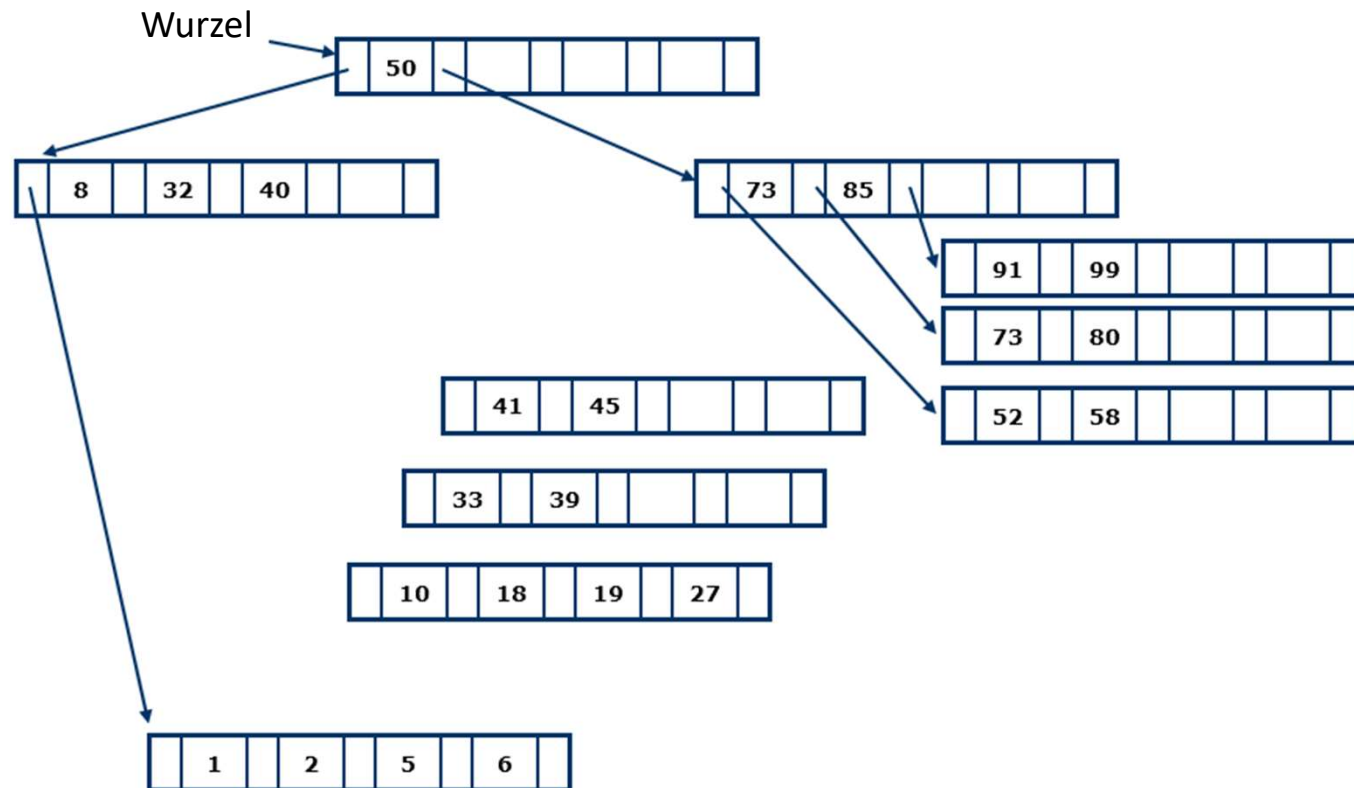
## Aufgabe 9 – lösche „8“ mit Merge-Operationen



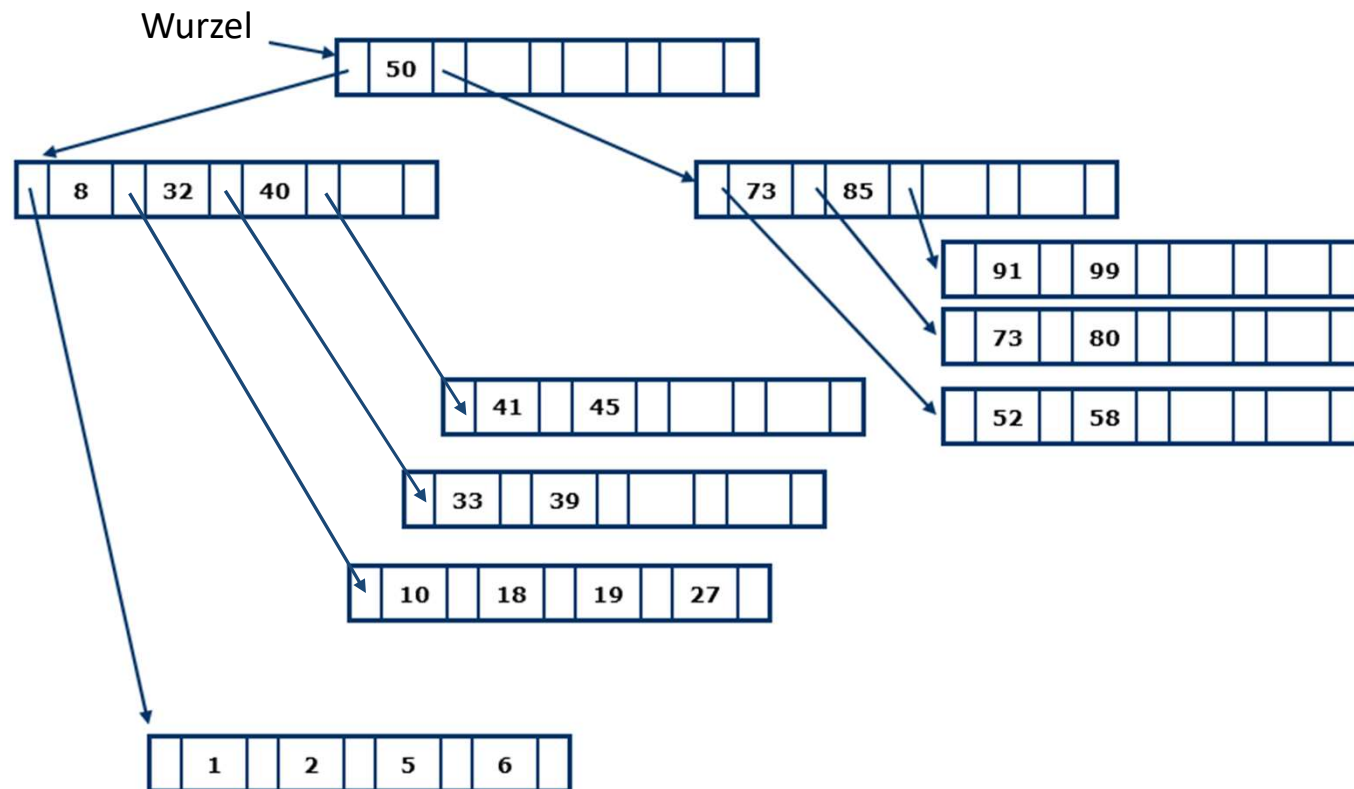
## Aufgabe 9 – lösche „8“ mit Merge-Operationen



## Aufgabe 9 – lösche „8“ mit Merge-Operationen



## Aufgabe 9 – lösche „8“ mit Merge-Operationen



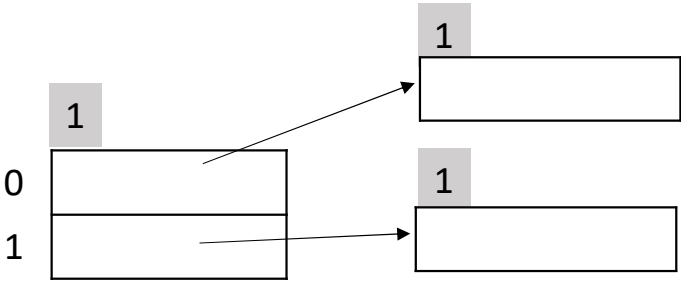
# Aufgabe 10

Wir benutzen erweiterbares Hashing für eine Datei, die Datensätze mit folgenden Suchschlüsselwerte enthält:

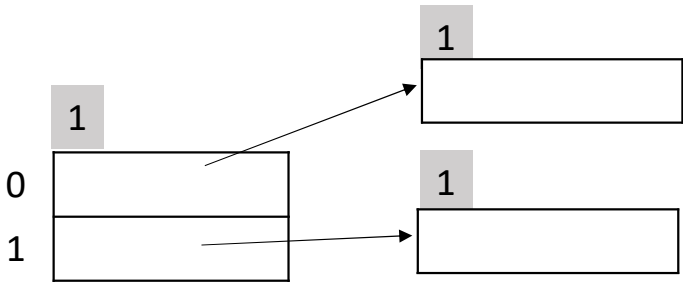
2, 3, 5, 7, 11, 17, 19, 23, 29

Zeige die Struktur der Hashdatei, wenn die Hashfunktion  $h(x) = x \bmod 8$  ist und die Behälter je 3 Einträge speichern können.

# Aufgabe 10

[illegible]

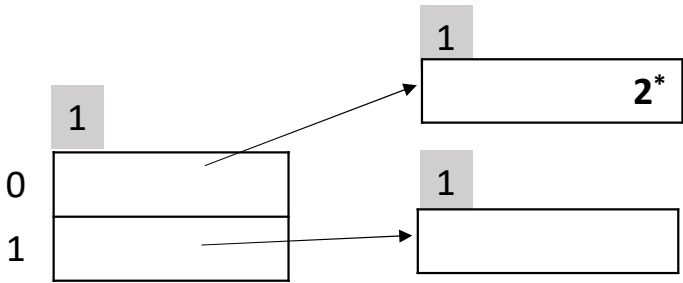
# Aufgabe 10



x	X mod 8	binär
2	2	010

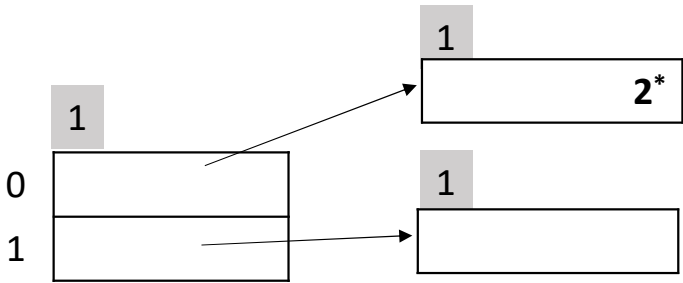


# Aufgabe 10



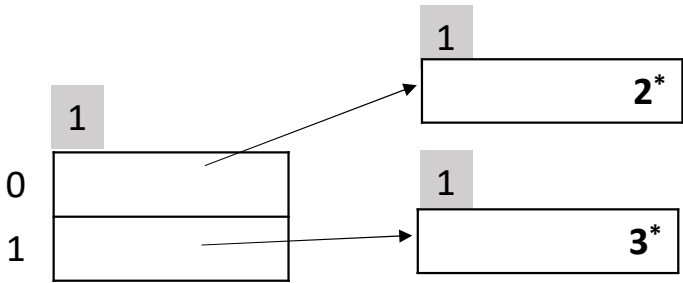
x	X mod 8	binär
2	2	010

# Aufgabe 10



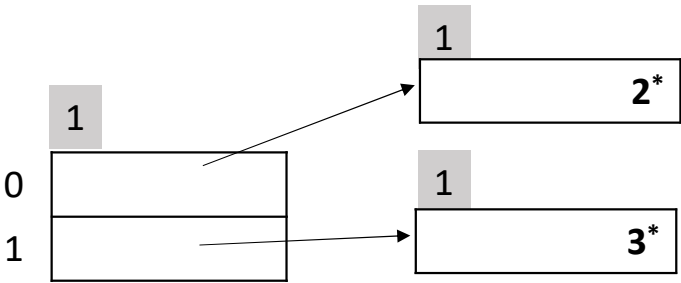
x	X mod 8	binär
2	2	010
3	3	011

# Aufgabe 10



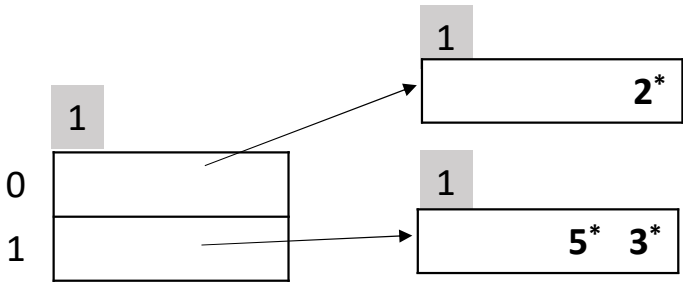
x	X mod 8	binär
2	2	010
3	3	011

# Aufgabe 10



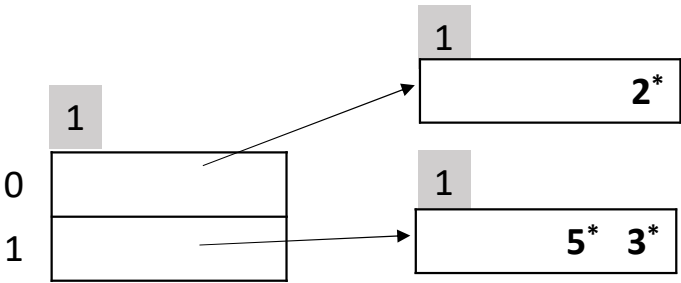
x	X mod 8	binär
2	2	010
3	3	011
5	5	101

# Aufgabe 10



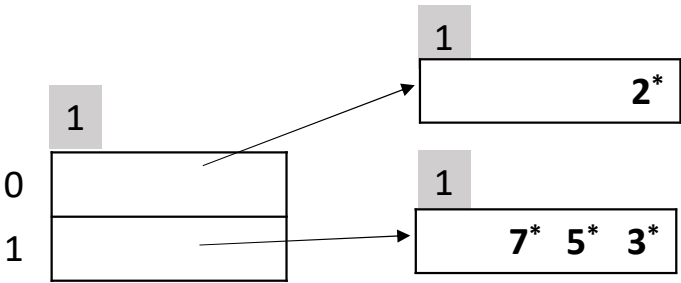
x	X mod 8	binär
2	2	010
3	3	011
5	5	101

# Aufgabe 10



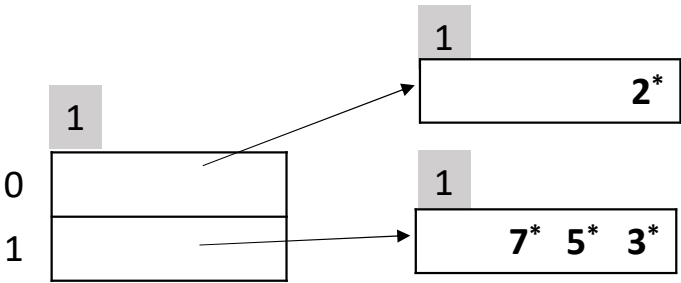
x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111

# Aufgabe 10



x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111

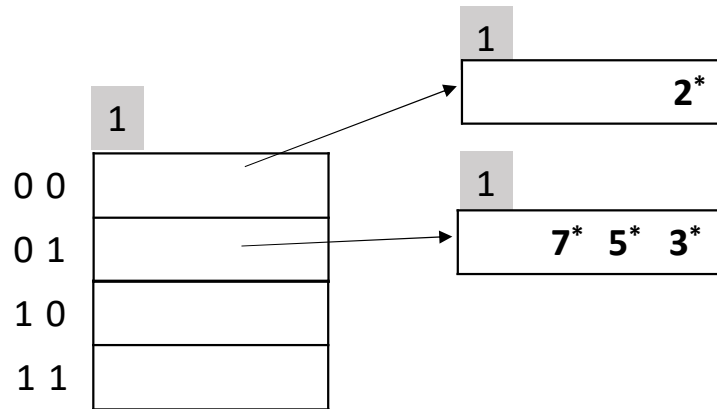
# Aufgabe 10



x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011

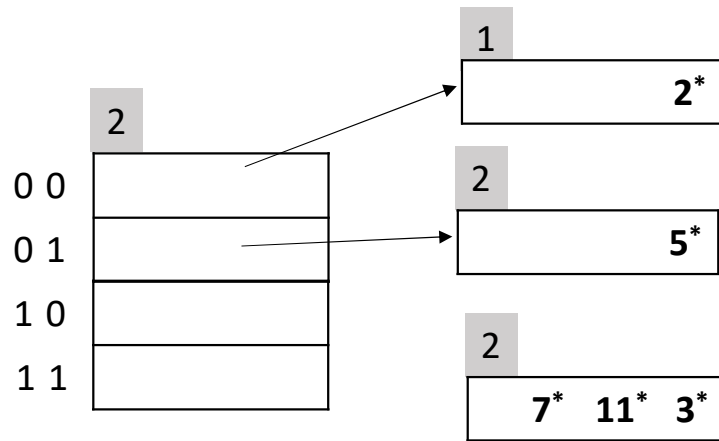


# Aufgabe 10



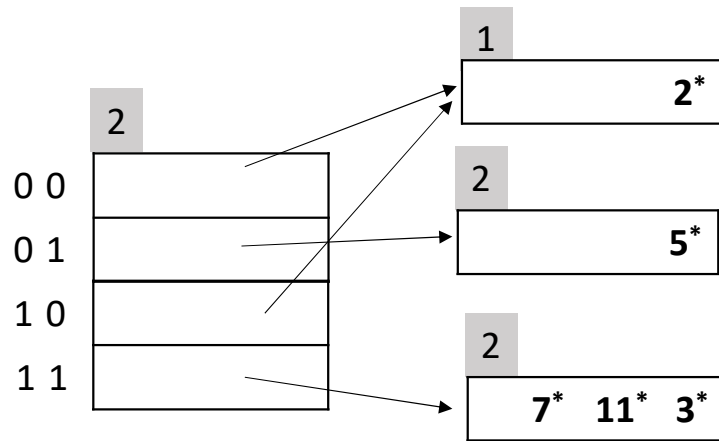
x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011

# Aufgabe 10



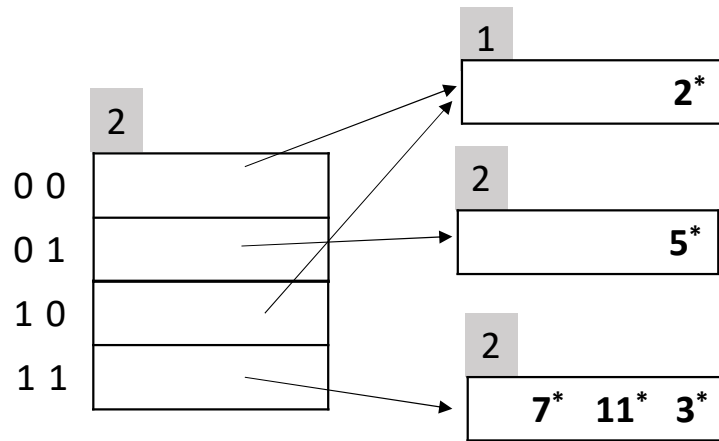
x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011

# Aufgabe 10



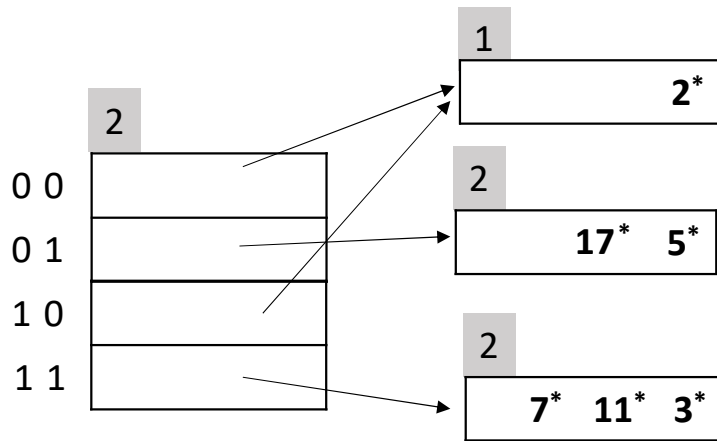
x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011

# Aufgabe 10



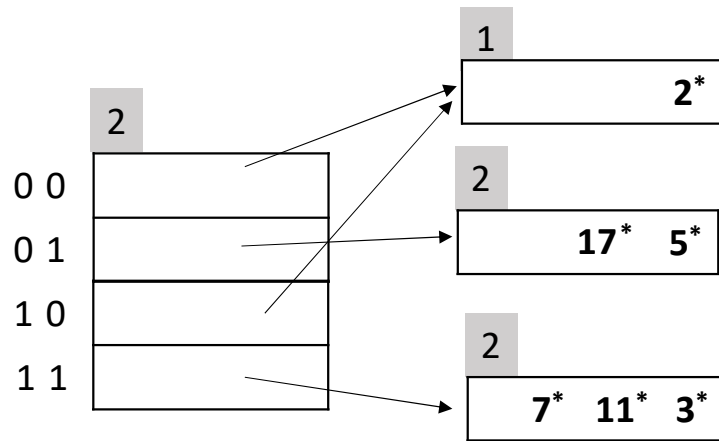
x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011
17	1	001

# Aufgabe 10



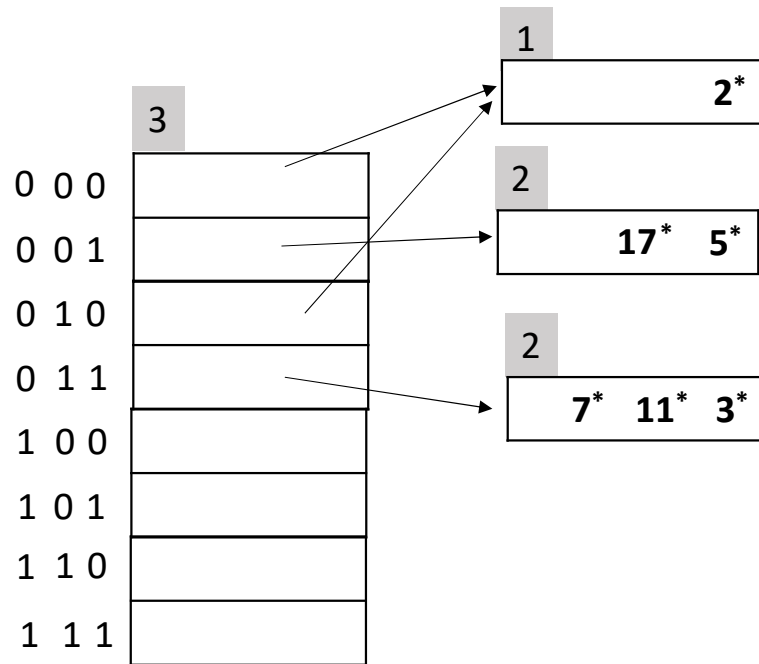
x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011
17	1	001

# Aufgabe 10



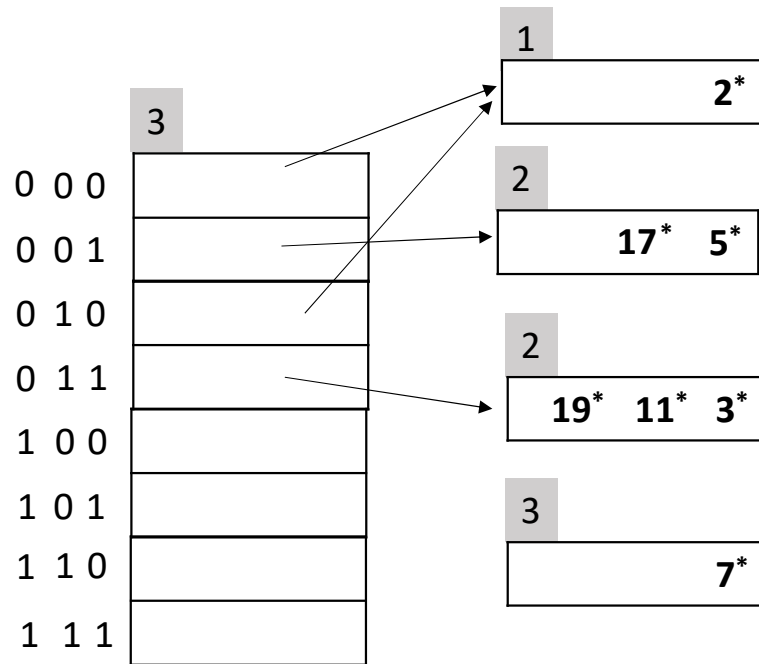
x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011
17	1	001
19	3	011

# Aufgabe 10



x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011
17	1	001
19	3	011

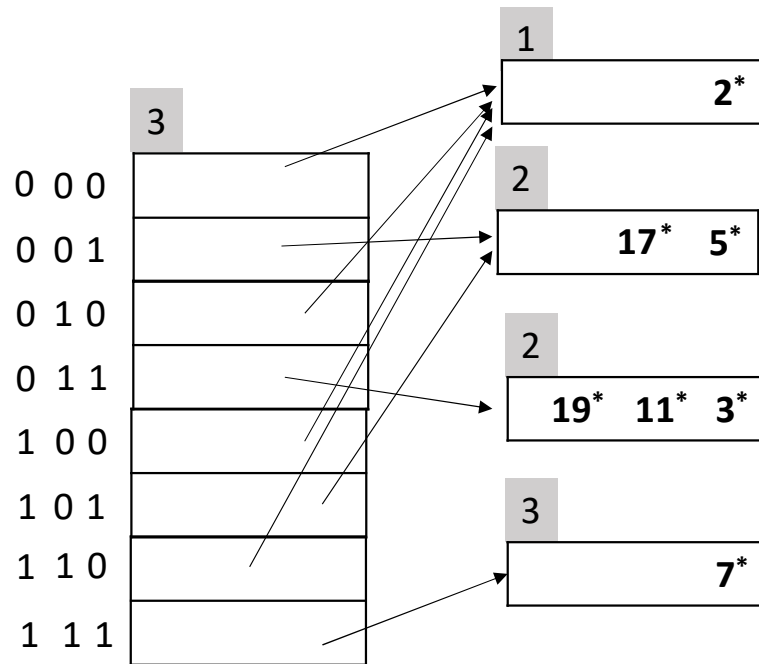
# Aufgabe 10



x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011
17	1	001
19	3	011

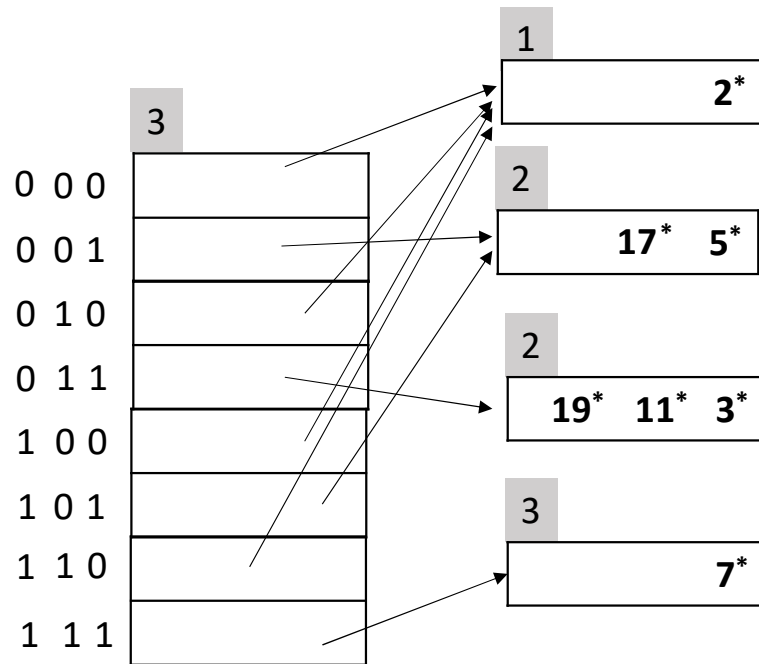


# Aufgabe 10



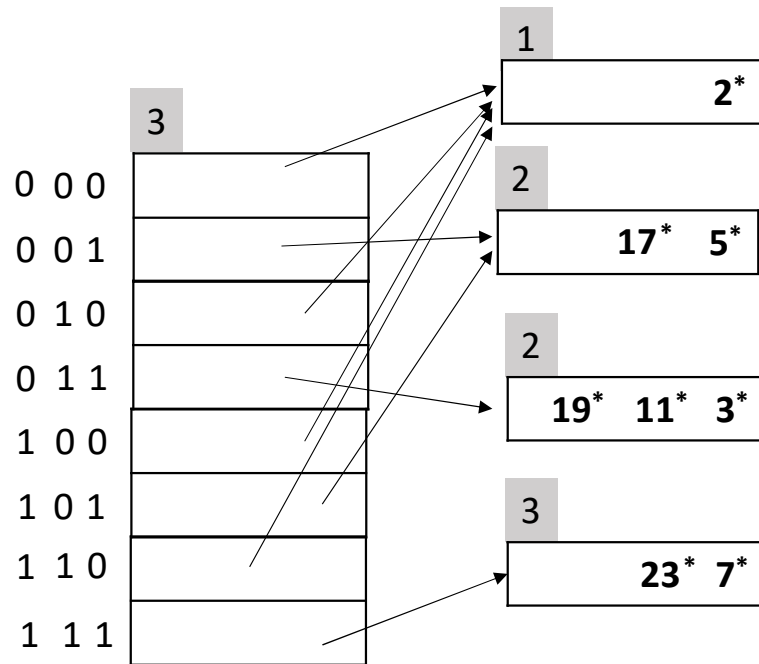
x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011
17	1	001
19	3	011

# Aufgabe 10



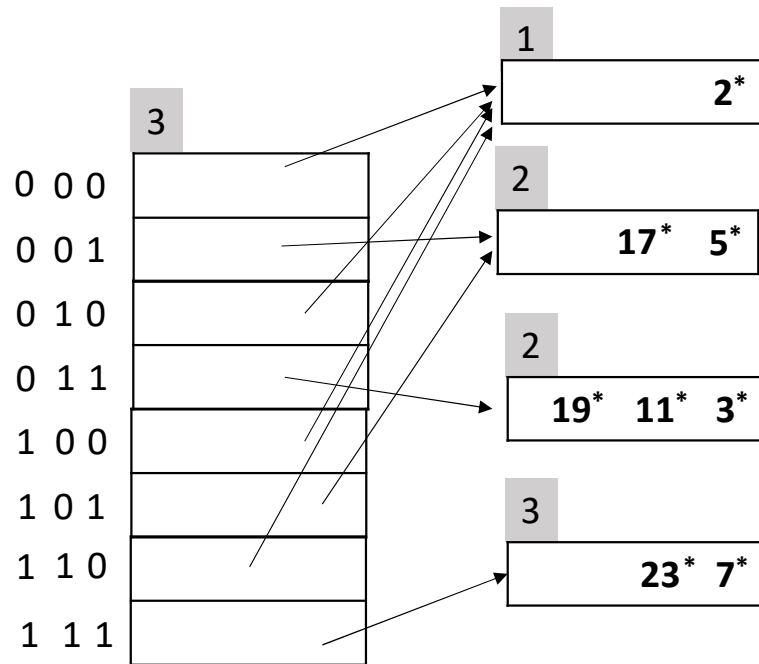
x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011
17	1	001
19	3	011
23	7	111

# Aufgabe 10



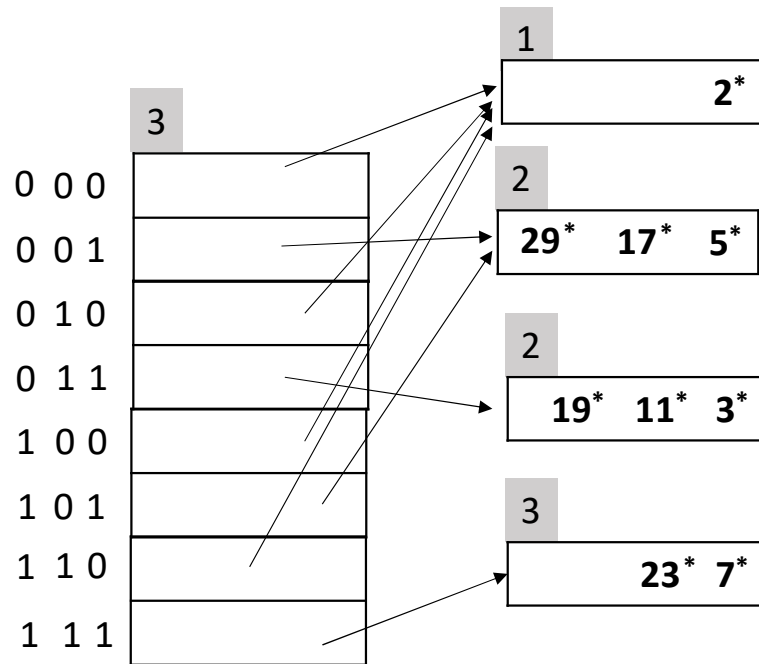
x	X mod 8	binär
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011
17	1	001
19	3	011
23	7	111

# Aufgabe 10



x	X mod 8	binär
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

# Aufgabe 10



x	X mod 8	binär
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