Datenstrukturen, Algorithmen und Programmierung 2 (DAP2)

Teile & Herrsche

Integer Multiplikation

- Problem: Multipliziere zwei n-Bit Integer
- Eingabe: Zwei n-Bit Integer X,Y
- Ausgabe: 2n-Bit Integer Z mit Z=XY

Annahmen:

- Wir können n-Bit Integer in $\Theta(n)$ (worst case) Zeit addieren
- Wir können n-Bit Integer in $\Theta(n+k)$ (worst case) Zeit mit 2^k multiplizieren

Teile & Herrsche

Laufzeit Schulmethode

- n Multiplikationen mit 2^k für ein $k \leq n$
- n-1 Additionen im worst-case:

$$\underbrace{11\ldots111}_{\text{n-Bit}} \cdot \underbrace{11\ldots111}_{\text{n-Bit}}$$

- Jede Addition $\Theta(n)$ Zeit
- Insgesamt $\Theta(n^2)$ Laufzeit

Teile & Herrsche

Laufzeit Schulmethode

- n Multiplikationen mit 2^k für ein $k \leq n$
- n-1 Additionen im worst-case:

$$\underbrace{11\ldots111}_{\text{n-Bit}} \cdot \underbrace{11\ldots111}_{\text{n-Bit}}$$

- Jede Addition $\Theta(n)$ Zeit
- Insgesamt $\Theta(n^2)$ Laufzeit

Bessere Laufzeit mit Teile & Herrsche?

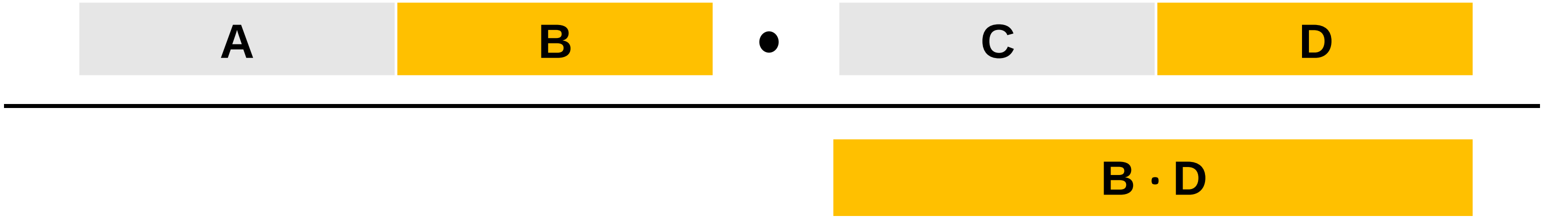
Teile & Herrsche

Integer Multiplikation

A	B	•	C	D

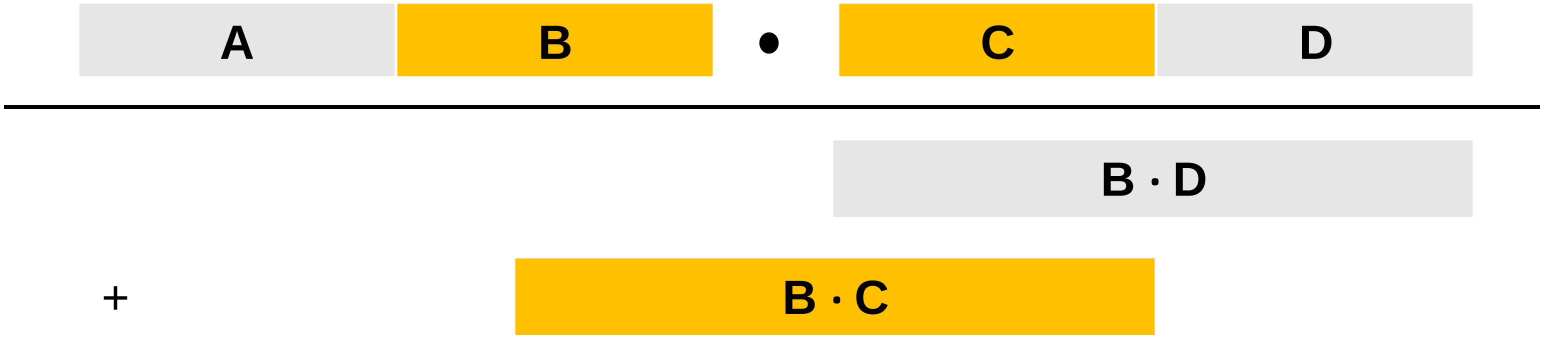
Teile & Herrsche

Integer Multiplikation



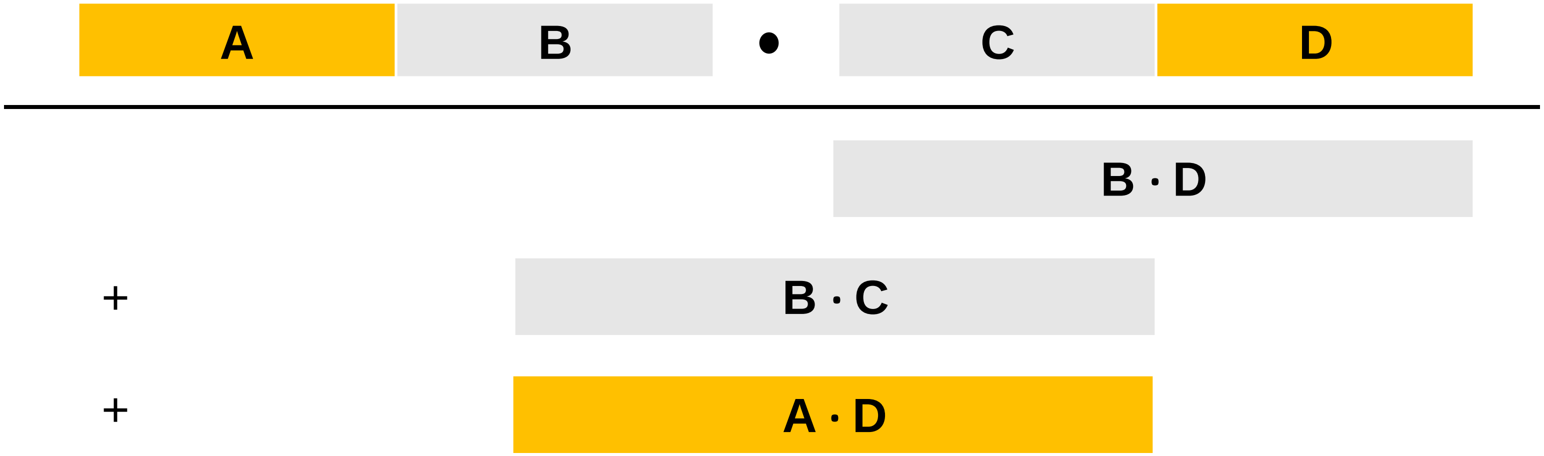
Teile & Herrsche

Integer Multiplikation



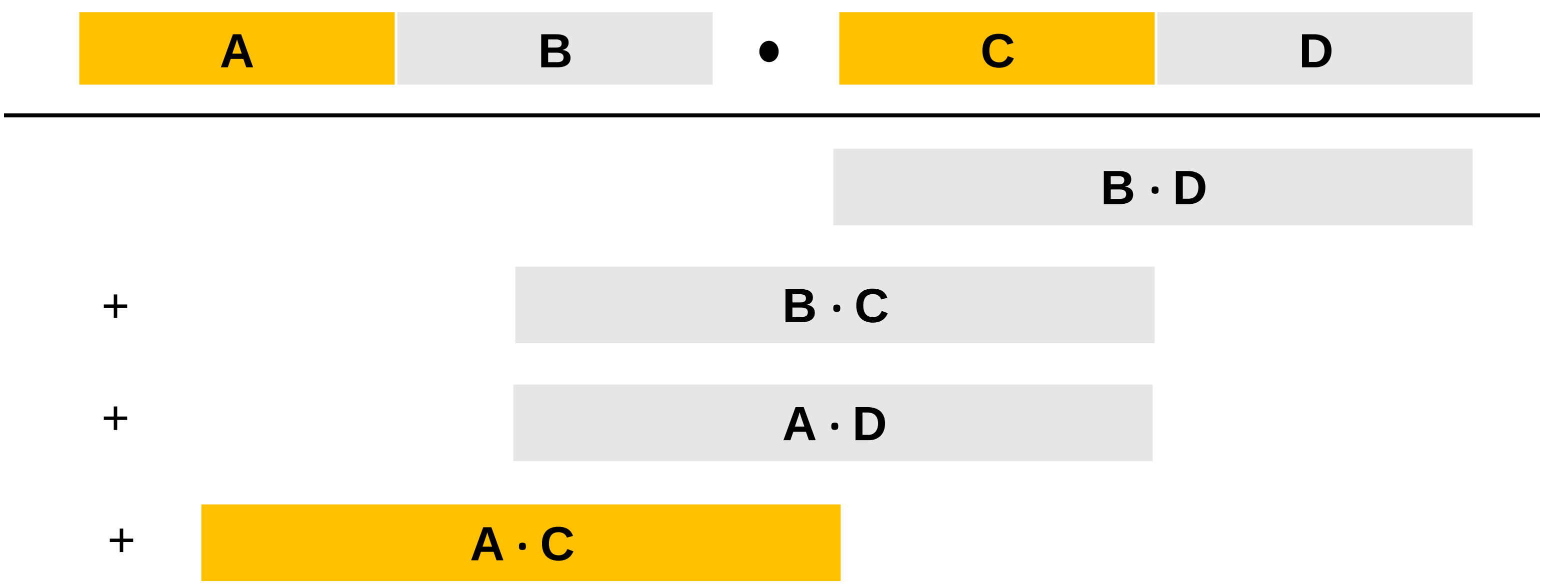
Teile & Herrsche

Integer Multiplikation



Teile & Herrsche

Integer Multiplikation



Teile & Herrsche

Integer Multiplikation

A **B** • **C** **D**

B · D

\+ **B · C**

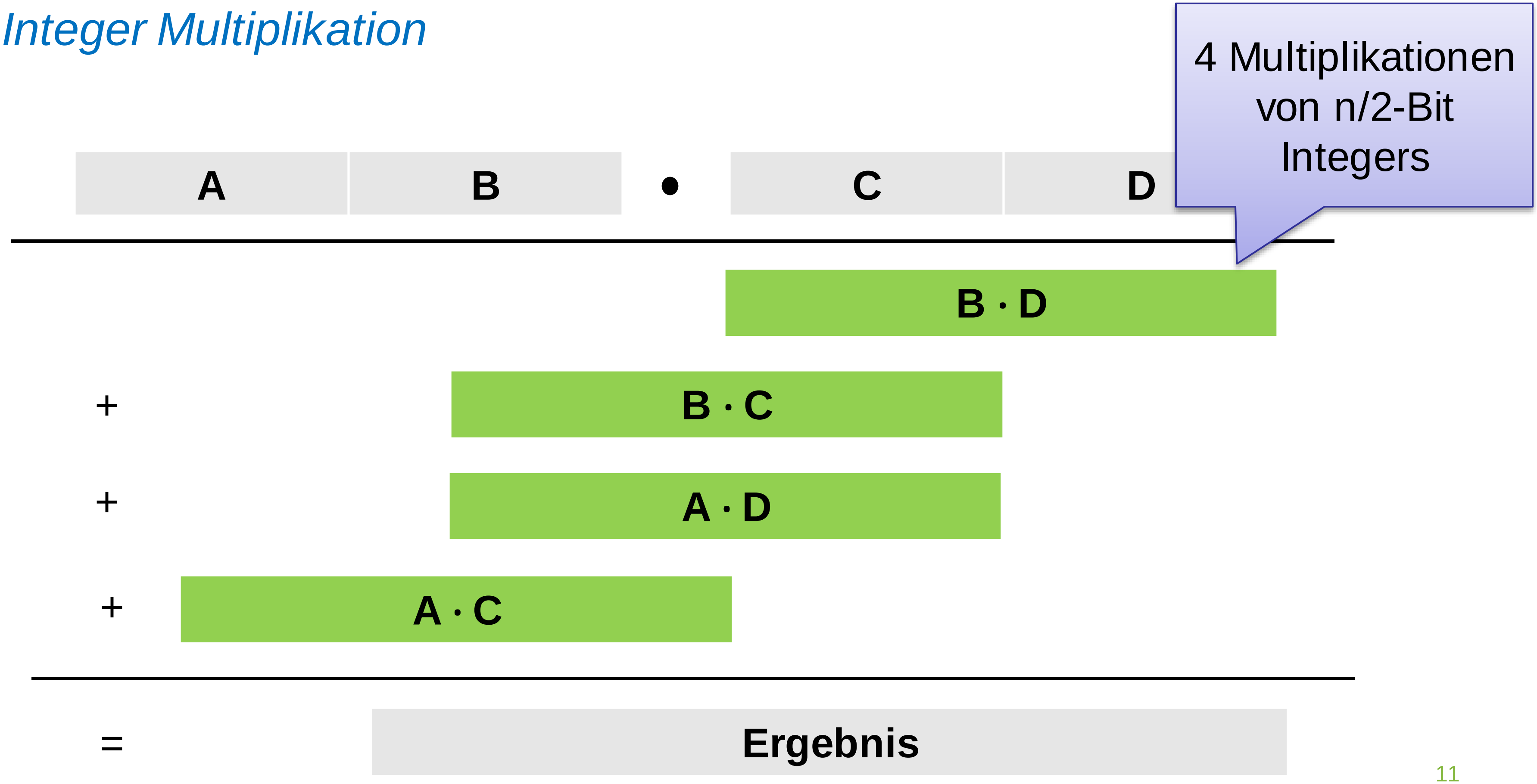
\+ **A · D**

\+ **A · C**

= **Ergebnis**

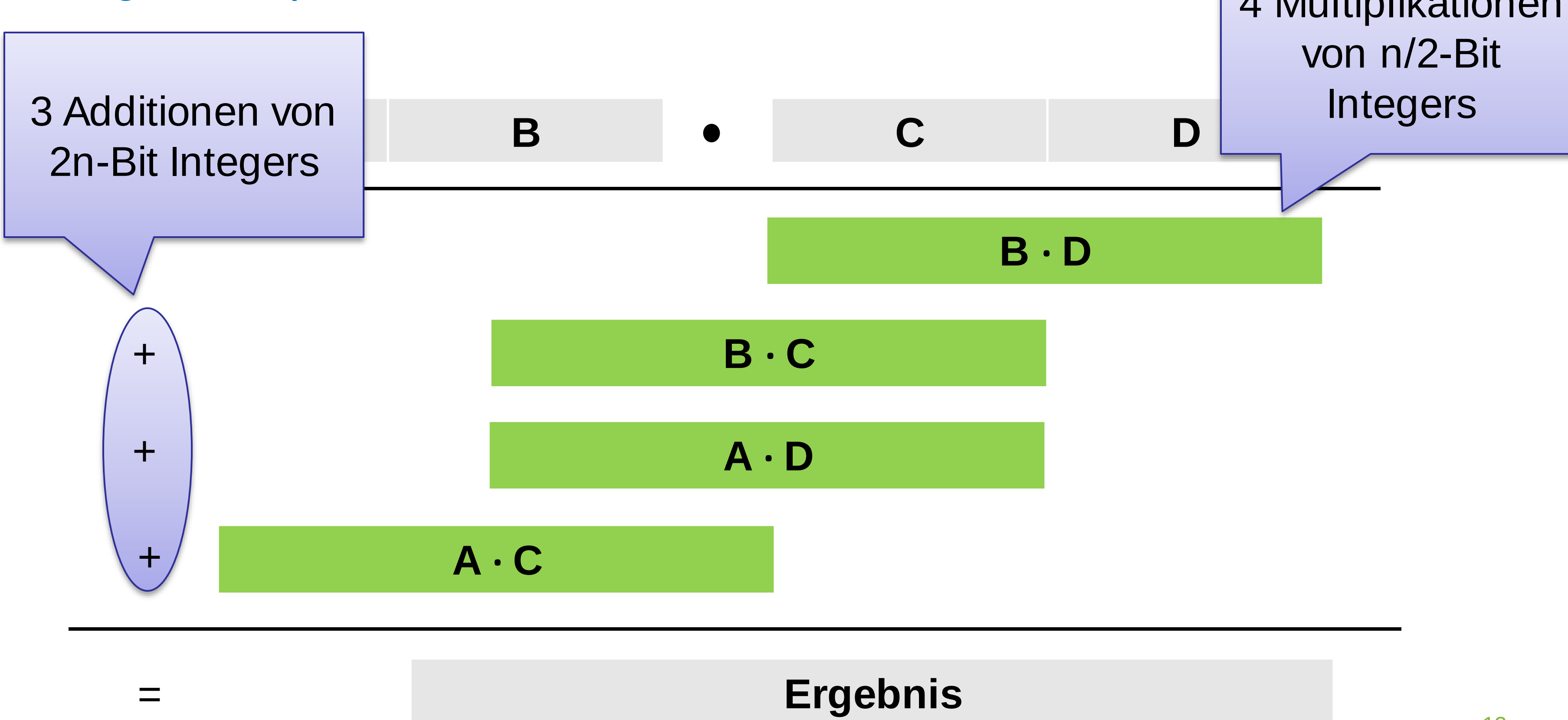
Teile & Herrsche

Integer Multiplikation



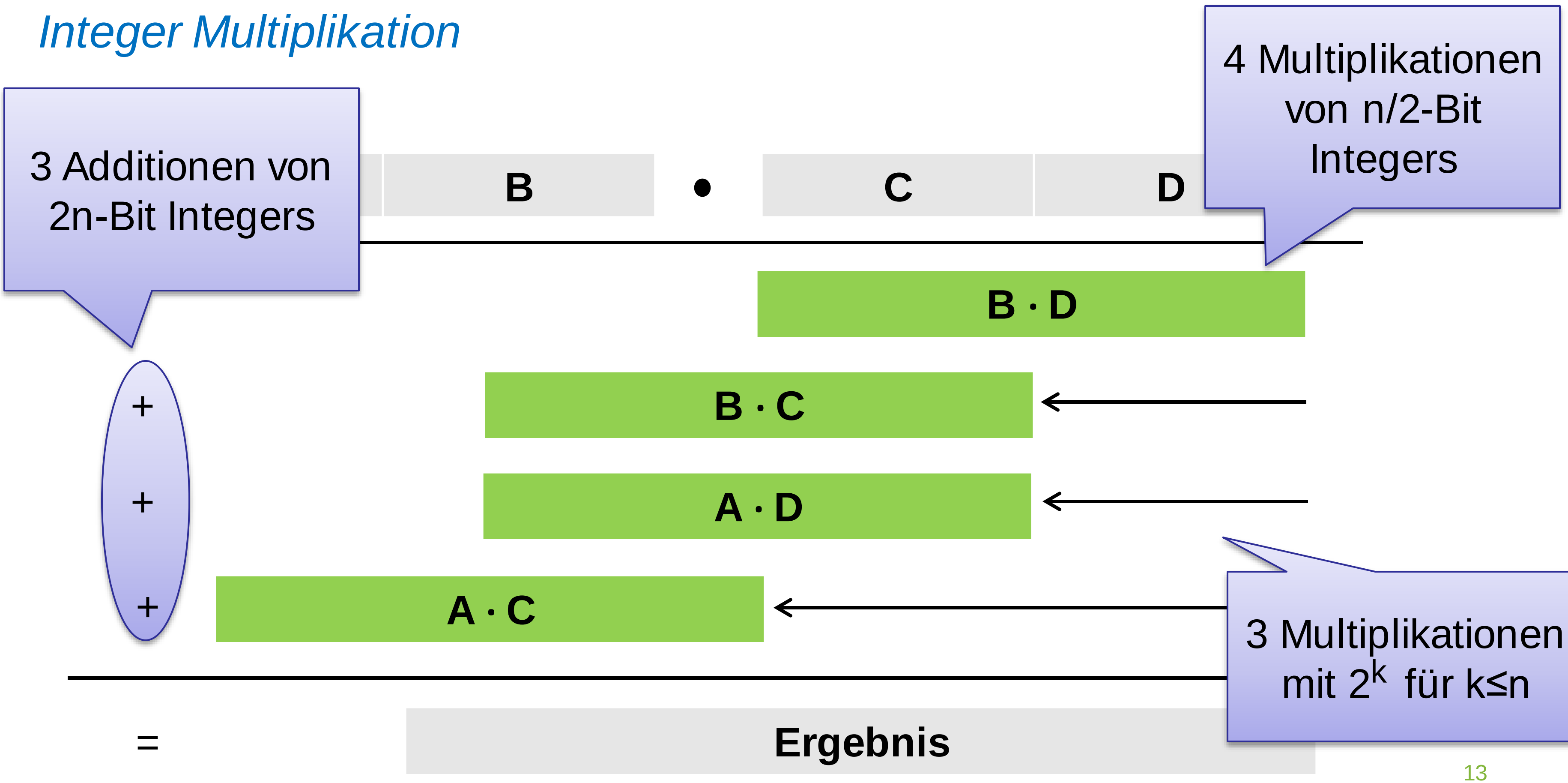
Teile & Herrsche

Integer Multiplikation



Teile & Herrsche

Integer Multiplikation



Teile & Herrsche

Beispiel Multiplikation Schulmethode

$$T(n) = 4 \cdot T(n/2) + cn$$

Anzahl Unterprobleme

Aufwand für Aufteilen und
Zusammenfügen

Größe der Unterprobleme
(bestimmt Höhe des Rekursionsbaums)

- (und T(1) = const)

(n Zweierpotenz)

Teile & Herrsche

Satz 8

Die Multiplikation zweier n-Bit Zahlen mit dem einfachen Teile & Herrsche Verfahren hat Laufzeit $O(n^2)$.

Beweis

- Wir nehmen an, dass n eine Zweierpotenz ist. Induktion über n. Wir zeigen $T(n) \le cn^2$.

Teile & Herrsche

Satz 8

Die Multiplikation zweier n-Bit Zahlen mit dem einfachen Teile & Herrsche Verfahren hat Laufzeit $O(n^2)$.

Beweis

- Wir nehmen an, dass n eine Zweierpotenz ist. Induktion über n. Wir zeigen $T(n) \leq cn^2$.
- (I.A.) Die Laufzeit zur Multiplikation von zwei 1-Bit Zahlen ist höchstens c.

Teile & Herrsche

Satz 8

Die Multiplikation zweier n-Bit Zahlen mit dem einfachen Teile & Herrsche Verfahren hat Laufzeit $O(n^2)$.

Beweis

- Wir nehmen an, dass n eine Zweierpotenz ist. Induktion über n. Wir zeigen $T(n) \le cn^2$.
- (I.A.) Die Laufzeit zur Multiplikation von zwei 1-Bit Zahlen ist höchstens c.
- (I.V.) Für jedes $m<n$ ist die Laufzeit zur Multiplikation von zwei m-Bit Zahlen cm^2.

Teile & Herrsche

Satz 8

Die Multiplikation zweier n-Bit Zahlen mit dem einfachen Teile & Herrsche Verfahren hat Laufzeit $O(n^2)$.

Beweis

- Wir nehmen an, dass n eine Zweierpotenz ist. Induktion über n. Wir zeigen $T(n) \le cn^2$.
- (I.A.) Die Laufzeit zur Multiplikation von zwei 1-Bit Zahlen ist höchstens c.
- (I.V.) Für jedes $m<n$ ist die Laufzeit zur Multiplikation von zwei m-Bit Zahlen cm^2.
- (I.S.) Betrachte eine Multiplikation von zwei n-Bit Zahlen. Es gilt $T(n) \le 4\,T(n/2) + cn$. Nach (I.V.) gilt dann $T(n) \le 4\,c\,(n/2)^2 + cn = cn^2+cn$.

Teile & Herrsche

Satz 8

Die Multiplikation zweier n-Bit Zahlen mit dem einfachen Teile & Herrsche Verfahren hat Laufzeit $O(n^2)$.

Beweis

- Wir nehmen an, dass n eine Zweierpotenz ist. Induktion über n. Wir zeigen $T(n) \le cn^2$.
- (I.A.) Die Laufzeit zur Multiplikation von zwei 1-Bit Zahlen ist höchstens c.
- (I.V.) Für jedes m<n ist die Laufzeit zur Multiplikation von zwei m-Bit Zahlen cm^2.
- (I.S.) Betrachte eine Multiplikation von zwei n-Bit Zahlen. Es gilt $T(n) \le 4\, T(n/2) + cn$. Nach (I.V.) gilt dann $T(n) \le 4\, c\, (n/2)^2 + cn = \boxed{cn^2+cn.}$

Funktioniert nicht !!!!!

Teile & Herrsche

Satz 8

Die Multiplikation zweier n-Bit Zahlen mit dem einfachen Teile & Herrsche Verfahren hat Laufzeit $O(n^2)$.

Beweis (neuer Versuch)

- Wir nehmen an, dass n eine Zweierpotenz ist. Induktion über n. O.b.d.A. sei $c \geq T(2)$. Wir zeigen $\boxed{T(n) \leq cn^2 - cn.}$

Trick: Die Funktion etwas verkleinern!!

Teile & Herrsche

Satz 8

Die Multiplikation zweier n-Bit Zahlen mit dem einfachen Teile & Herrsche Verfahren hat Laufzeit $O(n^2)$.

Beweis (neuer Versuch)

- Wir nehmen an, dass n eine Zweierpotenz ist. Induktion über n. O.b.d.A. sei $c \geq T(2)$. Wir zeigen $T(n) \leq cn^2 - cn$.
- (I.A.) Die Laufzeit zur Multiplikation von zwei 2-Bit Zahlen ist höchstens $T(2) \leq c \leq 2c$.

Teile & Herrsche

Satz 8

Die Multiplikation zweier n-Bit Zahlen mit dem einfachen Teile & Herrsche Verfahren hat Laufzeit $O(n^2)$.

Beweis (neuer Versuch)

- Wir nehmen an, dass n eine Zweierpotenz ist. Induktion über n. O.b.d.A. sei $c \geq T(2)$. Wir zeigen $T(n) \leq cn^2 - cn$.
- (I.A.) Die Laufzeit zur Multiplikation von zwei 2-Bit Zahlen ist höchstens $T(2) \leq c \leq 2c$.
- (I.V.) Für jedes $m<n$ ist die Laufzeit zur Multiplikation von zwei m-Bit Zahlen $c\,m^2 - cm$.

Teile & Herrsche

Satz 8

Die Multiplikation zweier n-Bit Zahlen mit dem einfachen Teile & Herrsche Verfahren hat Laufzeit $O(n^2)$.

Beweis (neuer Versuch)

- Wir nehmen an, dass n eine Zweierpotenz ist. Induktion über n. O.b.d.A. sei $c \geq T(2)$. Wir zeigen $T(n) \leq cn^2 - cn$.
- (I.A.) Die Laufzeit zur Multiplikation von zwei 2-Bit Zahlen ist höchstens $T(2) \leq c \leq 2c$.
- (I.V.) Für jedes $m<n$ ist die Laufzeit zur Multiplikation von zwei m-Bit Zahlen $c\,m^2 - cm$.
- (I.S.) Betrachte eine Multiplikation von zwei n-Bit Zahlen. Es gilt $T(n) \leq 4\,T(n/2) + cn$.

Teile & Herrsche

Satz 8

Die Multiplikation zweier n-Bit Zahlen mit dem einfachen Teile & Herrsche Verfahren hat Laufzeit $O(n^2)$.

Beweis (neuer Versuch)

- Wir nehmen an, dass n eine Zweierpotenz ist. Induktion über n. O.b.d.A. sei $c \geq T(2)$. Wir zeigen $T(n) \leq cn^2 - cn$.
- (I.A.) Die Laufzeit zur Multiplikation von zwei 2-Bit Zahlen ist höchstens $T(2) \leq c \leq 2c$.
- (I.V.) Für jedes $m<n$ ist die Laufzeit zur Multiplikation von zwei m-Bit Zahlen $c\, m^2 - cm$.
- (I.S.) Betrachte eine Multiplikation von zwei n-Bit Zahlen. Es gilt $T(n) \leq 4\, T(n/2) + cn$.
  Nach (I.V.) gilt dann $T(n) \leq 4\, c\, (n/2)^2 - 4\, c(n/2) + cn = cn^2 - cn$.

Teile & Herrsche

Satz 8

Die Multiplikation zweier n-Bit Zahlen mit dem einfachen Teile & Herrsche Verfahren hat Laufzeit $O(n^2)$.

Beweis (neuer Versuch)

- Wir nehmen an, dass n eine Zweierpotenz ist. Induktion über n. O.b.d.A. sei $c \geq T(2)$. Wir zeigen $T(n) \leq cn^2 - cn$.
- (I.A.) Die Laufzeit zur Multiplikation von zwei 2-Bit Zahlen ist höchstens $T(2) \leq c \leq 2c$.
- (I.V.) Für jedes $m<n$ ist die Laufzeit zur Multiplikation von zwei m-Bit Zahlen $c\,m^2 - cm$.
- (I.S.) Betrachte eine Multiplikation von zwei n-Bit Zahlen. Es gilt
  $T(n) \leq 4\,T(n/2) + cn$.
  Nach (I.V.) gilt dann $T(n) \leq 4\,c\,(n/2)^2 - 4\,c(n/2) + cn = cn^2 - cn$.

Teile & Herrsche

Verbesserte Integer Multiplikation

A B • C D

B · D

\+ B · C

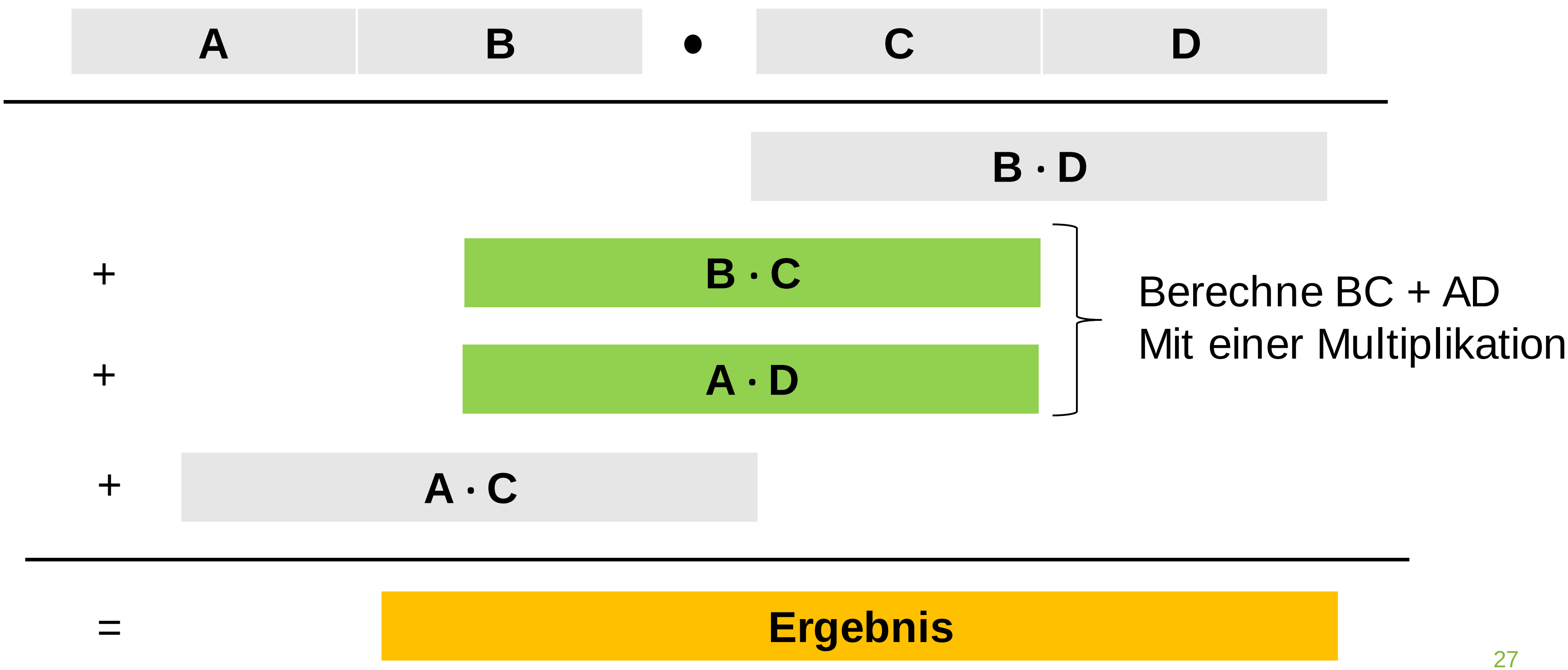
\+ A · D

\+ A · C

= **Ergebnis**

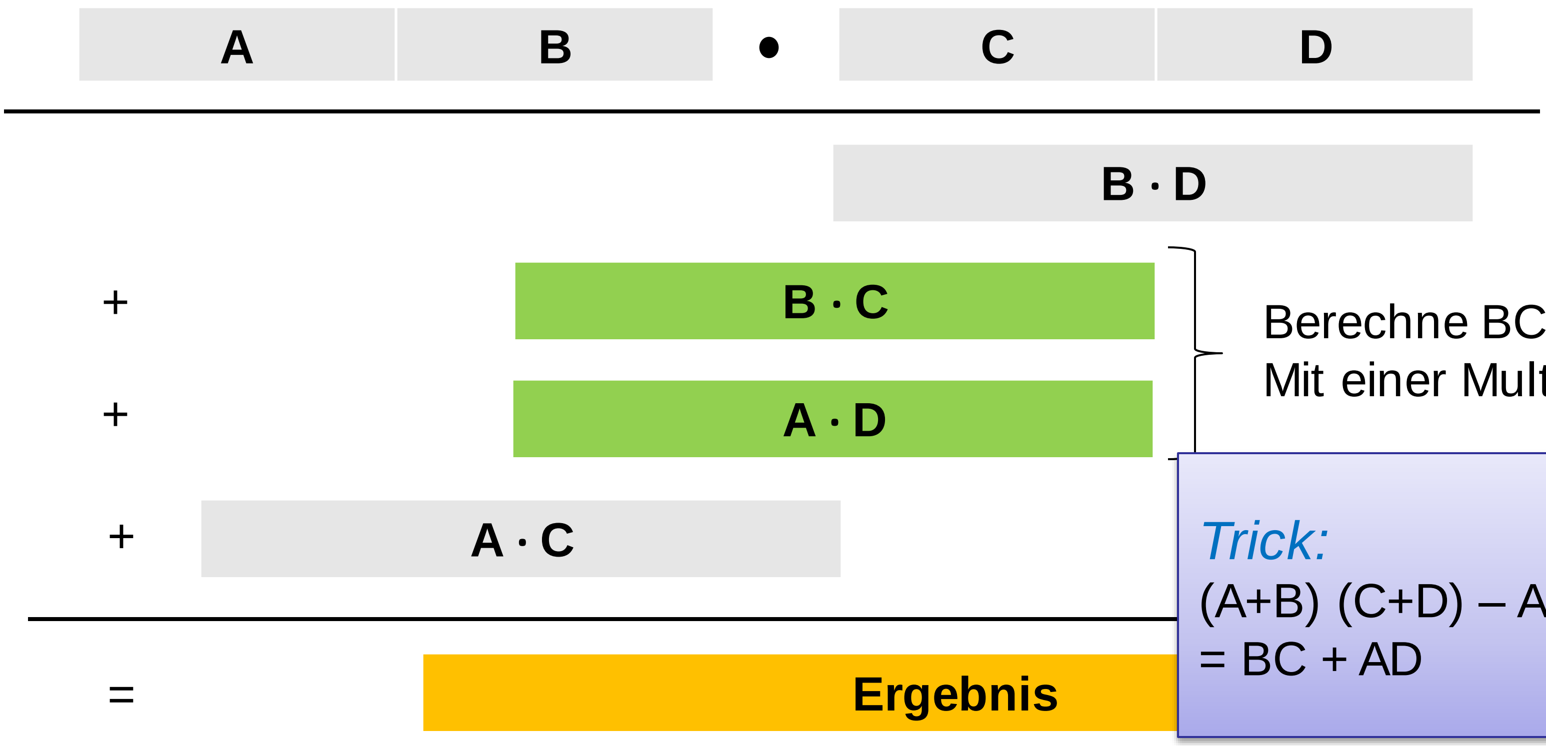
Teile & Herrsche

Verbesserte Integer Multiplikation



Teile & Herrsche

Verbesserte Integer Multiplikation



Berechne BC + AD
Mit einer Multiplikation

Trick:
(A+B) (C+D) – AC – BD
= BC + AD

Teile & Herrsche

Beispiel Schnelle Multiplikation

$$T(n) = 3 \cdot T(n/2) + cn$$

Anzahl Unterprobleme

Aufwand für Aufteilen und Zusammenfügen

Größe der Unterprobleme
(bestimmt Höhe des Rekursionsbaums)

- (und T(1) = const)

(n Zweierpotenz)

Teile & Herrsche

Aufwand Verbesserte Integer Multiplikation

- 3 Multiplikationen der Länge n/2
- [AC, BD, (A+B) (C+D)]
- Konstant viele Additionen und Multiplikationen mit Zweierpotenzen

Laufzeit

$$T(n) = \begin{cases} 3\,T(n/2) + cn & , n>1 \\ c & , n=1 \end{cases}$$

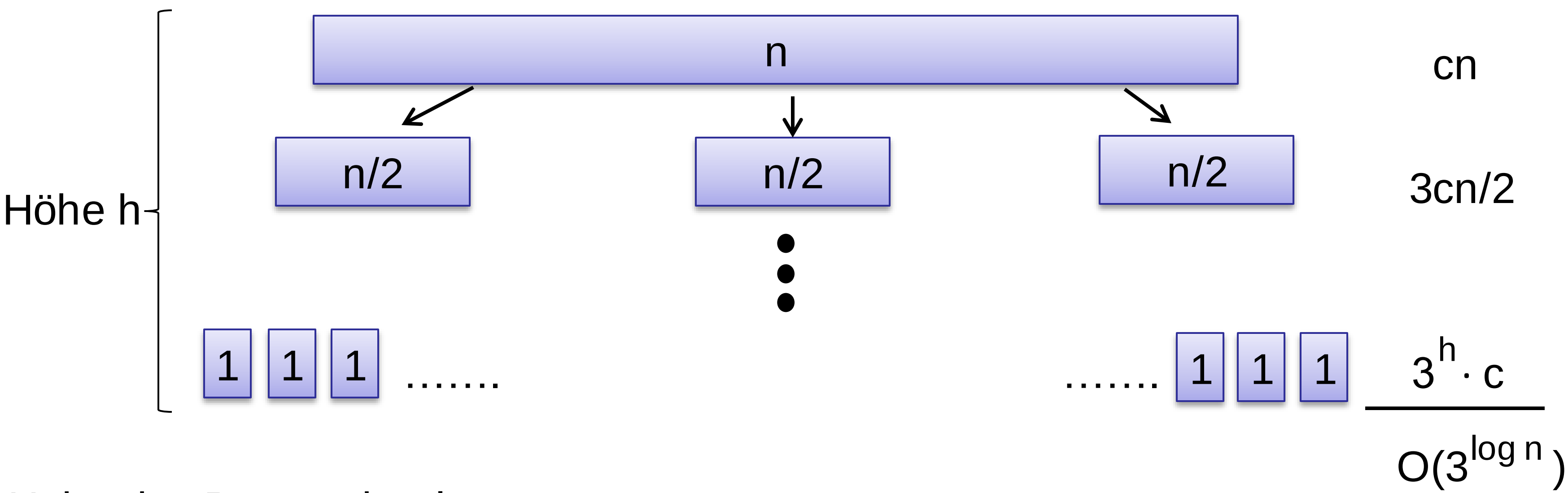
c geeignete Konstante

Teile & Herrsche

Laufzeit verbesserte Integer Multiplikation

$$T(n) \leq \begin{cases} 3\,T(n/2) + cn & , n>1 \\ c & , n=1 \end{cases}$$

c geeignete Konstante



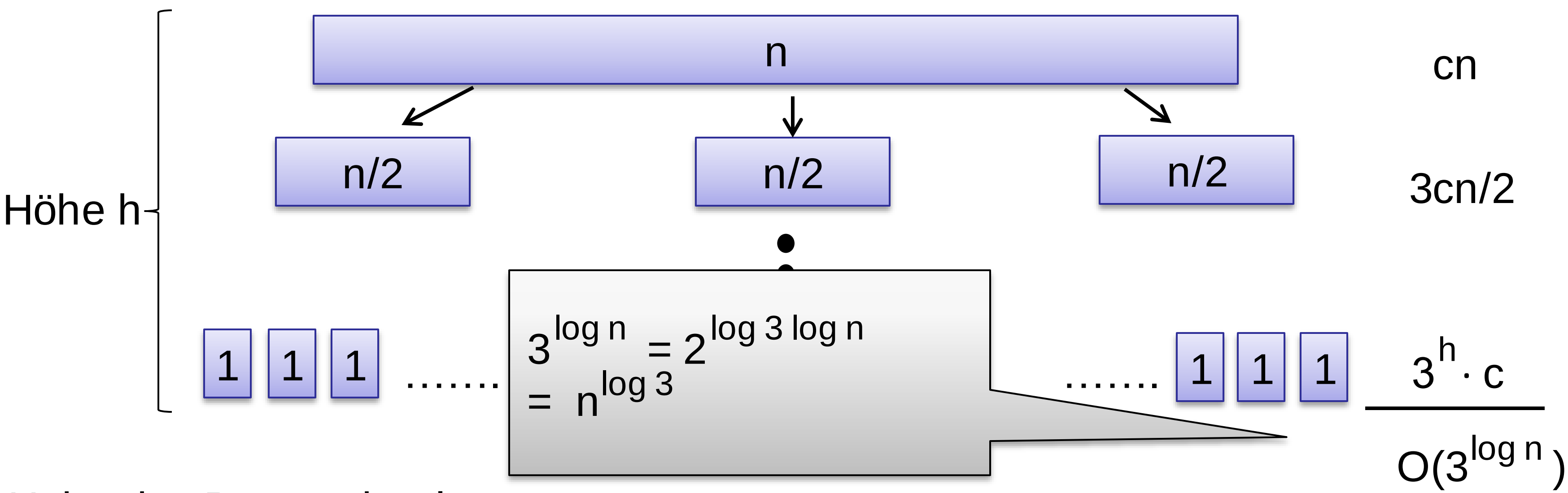
Höhe des Baums: h = log n

Teile & Herrsche

Laufzeit verbesserte Integer Multiplikation

$$T(n) \le \begin{cases} 3\,T(n/2) + cn & , n>1 \\ c & , n=1 \end{cases}$$

c geeignete Konstante



$3^{\log n} = 2^{\log 3 \log n}$
$= n^{\log 3}$

Höhe des Baums: h = log n

Teile & Herrsche

Satz 9

- Die Laufzeit der verbesserten Integer Multiplikation ist $O(3^{\log n}) = O(n^{\log 3})$.

Beweis

- Induktion über n. Sei $T(4) \le c$. Wir zeigen $T(n) \le c\, 3^{\log n} - 2cn$.

Teile & Herrsche

Satz 9

- Die Laufzeit der verbesserten Integer Multiplikation ist $O(3^{\log n}) = O(n^{\log 3})$.

Beweis

- Induktion über n. Sei $T(4) \le c$. Wir zeigen $T(n) \le c\, 3^{\log n} - 2cn$.
- (I.A.) Es gilt $T(4) \le c = c \cdot 3^{\log 4} - 2 \cdot c \cdot 4$.

Teile & Herrsche

Satz 9

- Die Laufzeit der verbesserten Integer Multiplikation ist $O(3^{\log n}) = O(n^{\log 3})$.

Beweis

- Induktion über n. Sei $T(4) \le c$. Wir zeigen $T(n) \le c\ 3^{\log n} - 2cn$.
- (I.A.) Es gilt $T(4) \le c = c \cdot 3^{\log 4} - 2 \cdot c \cdot 4$.
- (I.V.) Für jedes $m<n$ ist die Laufzeit für die Multiplikation zweier m-Bit Zahlen höchstens $c\ 3^{\log m} - 2cm$.

Teile & Herrsche

Satz 9

- Die Laufzeit der verbesserten Integer Multiplikation ist $O(3^{\log n}) = O(n^{\log 3})$.

Beweis

- Induktion über n. Sei $T(4) \le c$. Wir zeigen $T(n) \le c\, 3^{\log n} - 2cn$.
- (I.A.) Es gilt $T(4) \le c = c \cdot 3^{\log 4} - 2 \cdot c \cdot 4$.
- (I.V.) Für jedes m<n ist die Laufzeit für die Multiplikation zweier m-Bit Zahlen höchstens $c\, 3^{\log m} - 2cm$.
- (I.S.) Betrachte die Multiplikation von zwei n-Bit Zahlen. Es gilt $T(n) = 3\,T(n/2) + cn \le c \cdot 3^{\log n} - 6c\,(n/2) + cn = c \cdot 3^{\log n} - 2cn$.

Teile & Herrsche

Satz 9

- Die Laufzeit der verbesserten Integer Multiplikation ist $O(3^{\log n}) = O(n^{\log 3})$.

Beweis

- Induktion über n. Sei $T(4) \leq c$. Wir zeigen $T(n) \leq c\, 3^{\log n} - 2cn$.
- (I.A.) Es gilt $T(4) \leq c = c \cdot 3^{\log 4} - 2 \cdot c \cdot 4$.
- (I.V.) Für jedes $m<n$ ist die Laufzeit für die Multiplikation zweier m-Bit Zahlen höchstens $c\, 3^{\log m} - 2cm$.
- (I.S.) Betrachte die Multiplikation von zwei n-Bit Zahlen. Es gilt $T(n) = 3\, T(n/2) + cn \leq c \cdot 3^{\log n} - 6c\,(n/2) + cn = c \cdot 3^{\log n} - 2cn$.

Teile & Herrsche

Satz 10

- Zwei n-Bit Integer Zahlen können mit Hilfe des Teile & Herrsche Verfahrens in $O(n^{1.59})$ worst case Laufzeit multipliziert werden.

Beweis

- Die Laufzeit folgt aus Satz 12 wegen $1.59 \geq \log 3$. Korrektheit zeigen wir per Induktion über n.

Teile & Herrsche

Satz 10

- Zwei n-Bit Integer Zahlen können mit Hilfe des Teile & Herrsche Verfahrens in $O(n^{1.59})$ worst case Laufzeit multipliziert werden.

Beweis

- Die Laufzeit folgt aus Satz 12 wegen $1.59 \geq \log 3$. Korrektheit zeigen wir per Induktion über n.
- (I.A.) Die Multiplikation zweier 1-Bit Zahlen wird vom Rechner korrekt ausgeführt.

Teile & Herrsche

Satz 10

- Zwei n-Bit Integer Zahlen können mit Hilfe des Teile & Herrsche Verfahrens in $O(n^{1.59})$ worst case Laufzeit multipliziert werden.

Beweis

- Die Laufzeit folgt aus Satz 12 wegen $1.59 \geq \log 3$. Korrektheit zeigen wir per Induktion über n.
- (I.A.) Die Multiplikation zweier 1-Bit Zahlen wird vom Rechner korrekt ausgeführt.
- (I.V.) Die Multiplikation von zweier m-Bit Zahlen für m<n ist korrekt.

Teile & Herrsche

Satz 10

- Zwei n-Bit Integer Zahlen können mit Hilfe des Teile & Herrsche Verfahrens in $O(n^{1.59})$ worst case Laufzeit multipliziert werden.

Beweis

- Die Laufzeit folgt aus Satz 12 wegen $1.59 \geq \log 3$. Korrektheit zeigen wir per Induktion über n.
- (I.A.) Die Multiplikation zweier 1-Bit Zahlen wird vom Rechner korrekt ausgeführt.
- (I.V.) Die Multiplikation von zweier m-Bit Zahlen für $m<n$ ist korrekt.
- (I.S.) Nach (I.V.) werden die Produkte AC, BD, (A+B)(C+D) korrekt berechnet.

Teile & Herrsche

Satz 10

- Zwei n-Bit Integer Zahlen können mit Hilfe des Teile & Herrsche Verfahrens in $O(n^{1.59})$ worst case Laufzeit multipliziert werden.

Beweis

- Die Laufzeit folgt aus Satz 12 wegen $1.59 \geq \log 3$. Korrektheit zeigen wir per Induktion über n.
- (I.A.) Die Multiplikation zweier 1-Bit Zahlen wird vom Rechner korrekt ausgeführt.
- (I.V.) Die Multiplikation von zweier m-Bit Zahlen für $m<n$ ist korrekt.
- (I.S.) Nach (I.V.) werden die Produkte AC, BD, (A+B)(C+D) korrekt berechnet. Damit folgt die Korrektheit des Algorithmus wegen $(A+B)(C+D) - AC - BD = BC + AD$ und aufgrund unserer Vorüberlegungen.

Teile & Herrsche

Satz 10

- Zwei n-Bit Integer Zahlen können mit Hilfe des Teile & Herrsche Verfahrens in $O(n^{1.59})$ worst case Laufzeit multipliziert werden.

Beweis

- Die Laufzeit folgt aus Satz 12 wegen $1.59 \geq \log 3$. Korrektheit zeigen wir per Induktion über n.
- (I.A.) Die Multiplikation zweier 1-Bit Zahlen wird vom Rechner korrekt ausgeführt.
- (I.V.) Die Multiplikation von zweier m-Bit Zahlen für $m<n$ ist korrekt.
- (I.S.) Nach (I.V.) werden die Produkte AC, BD, (A+B)(C+D) korrekt berechnet. Damit folgt die Korrektheit des Algorithmus wegen $(A+B)(C+D) - AC - BD = BC + AD$ und aufgrund unserer Vorüberlegungen.

Teile & Herrsche

Laufzeiten der Form

$$T(n) = a \cdot T(n/b) + f(n)$$

Anzahl Unterprobleme

Aufwand für das Aufteilen und Zusammensetzen

Größe der Unterprobleme (bestimmt Höhe des Rekursionsbaums)

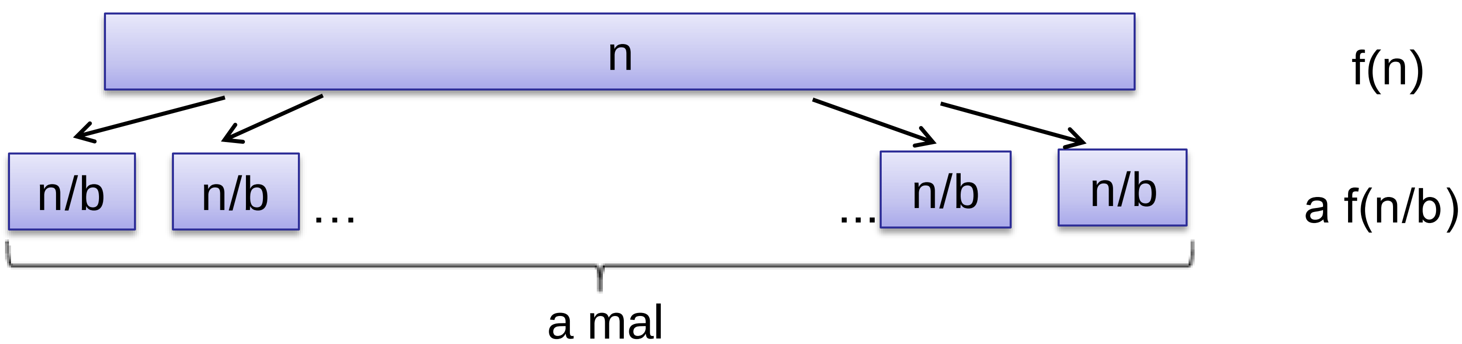
- wobei T(1) konstant ist

Teile & Herrsche

Laufzeit der Form

$$T(n) = \begin{cases} a\,T(n/b) + f(n) & , n>1 \\ 1 & , n=1 \end{cases}$$

$f(n) = O(n^k)$ für Konstante k.

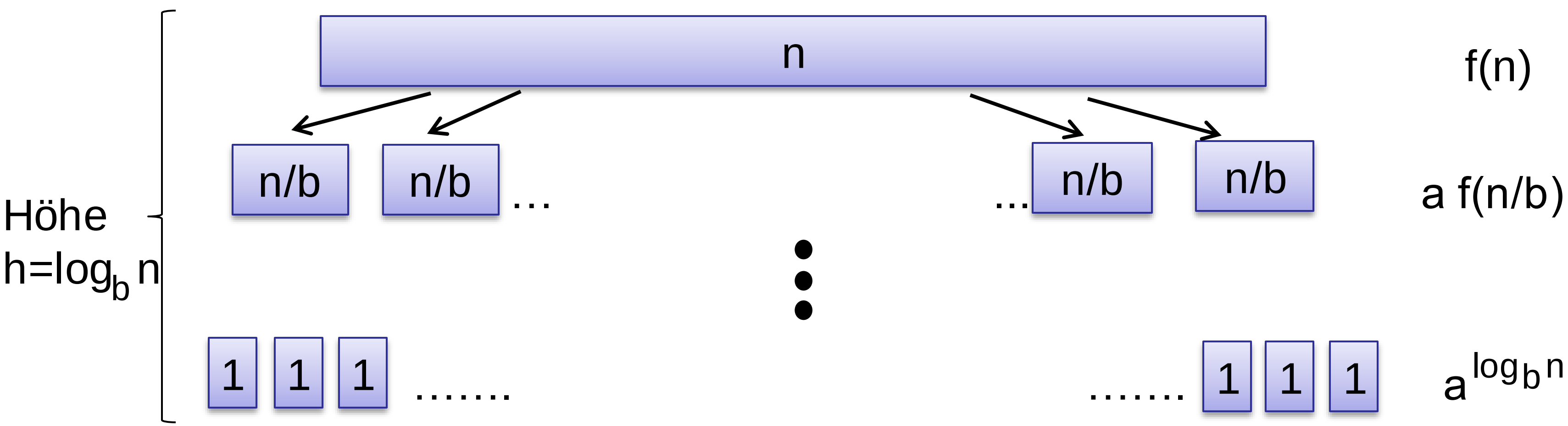


Teile & Herrsche

$f(n) = O(n^k)$ für Konstante k.

Laufzeit der Form

- Setze $\alpha \cdot f(n) = a \cdot f(n/b)$



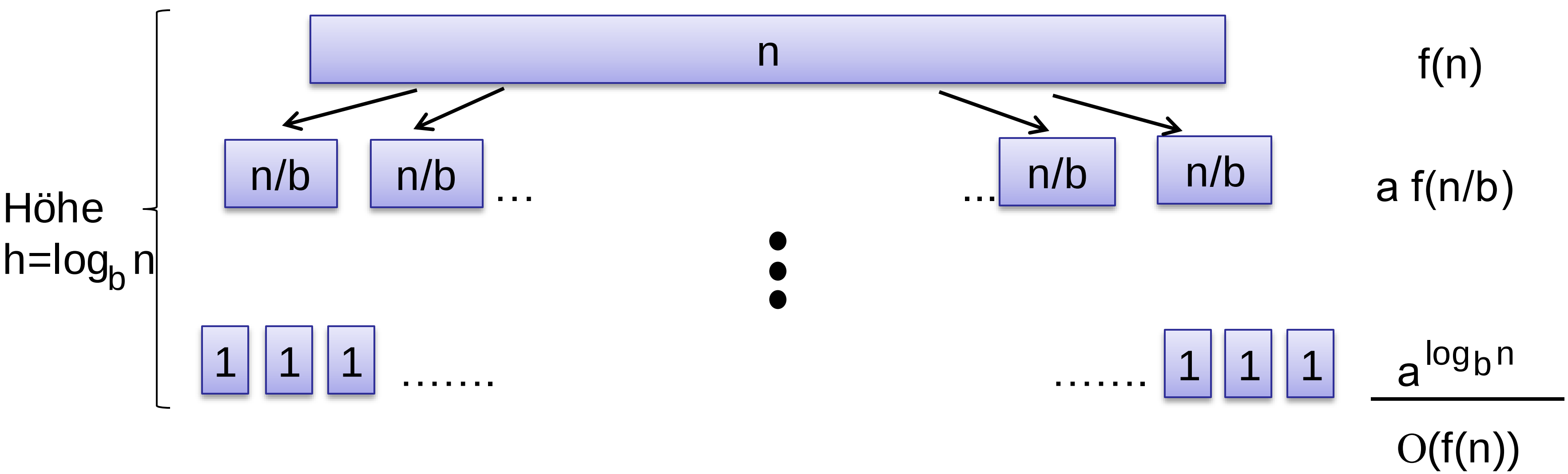
Teile & Herrsche

f(n) = O(n^k) für Konstante k.

Laufzeit der Form

- Setze $\alpha \cdot f(n) = a \cdot f(n/b)$

Fall 1:
α<1



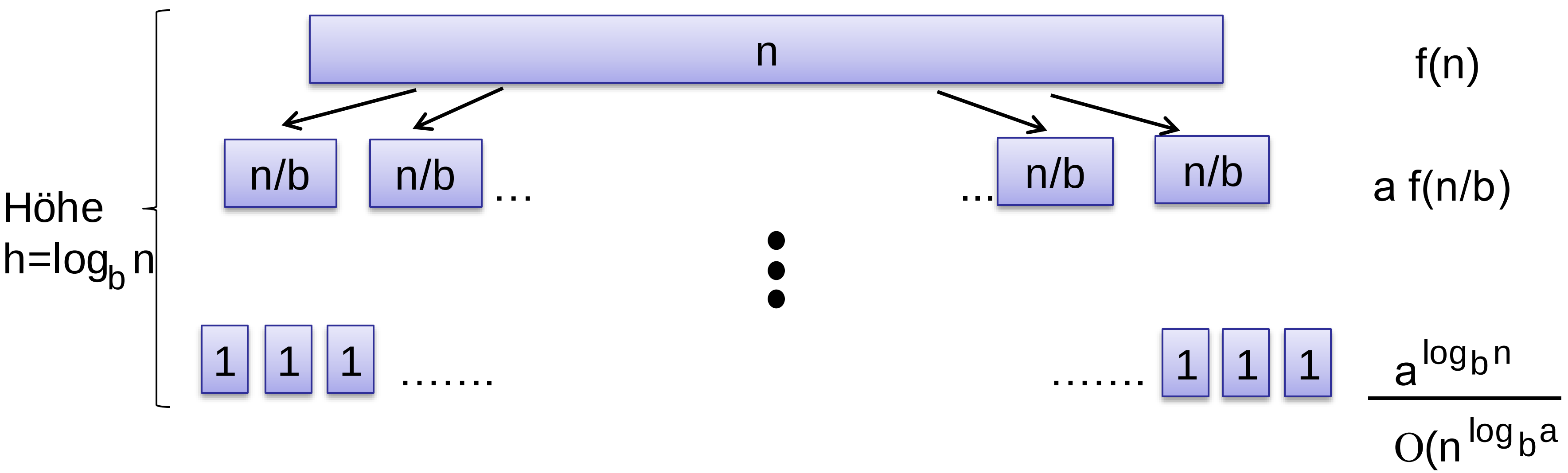
Teile & Herrsche

$f(n) = O(n^k)$ für Konstante k.

Fall 1:
$\alpha > 1$

Laufzeit der Form

- Setze $\alpha \cdot f(n) = a \cdot f(n/b)$



Teile & Herrsche

Laufzeit der Form

- Setze $\alpha \cdot f(n) = a \cdot f(n/b)$

$f(n) = O(n^k)$ für Konstante k.

Fall 1:
$\alpha=1$