eine

Schnuppervorlesung Informatik

ein bisschen Kombinatorik
ein Ausschnitt aus *Grundbegriffe der Informatik*

Thomas Worsch

KIT, Institut für Theoretische Informatik

Februar 2017

Der Plan

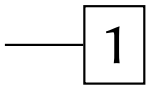
Binomialkoeffizienten

Huffman-Codierung

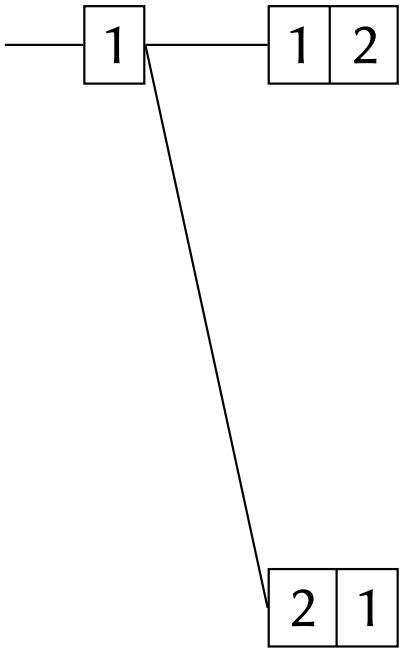
Ein bisschen Kombinatorik

Binomialkoeffizienten

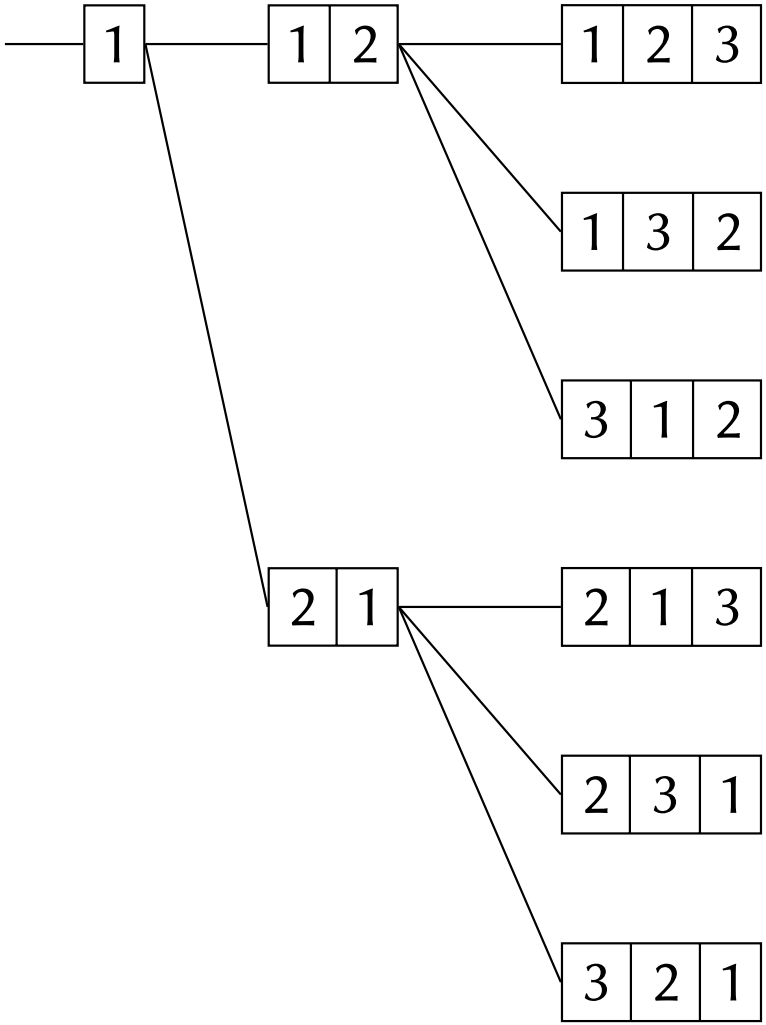
Unterscheidbare Kisten nebeneinander



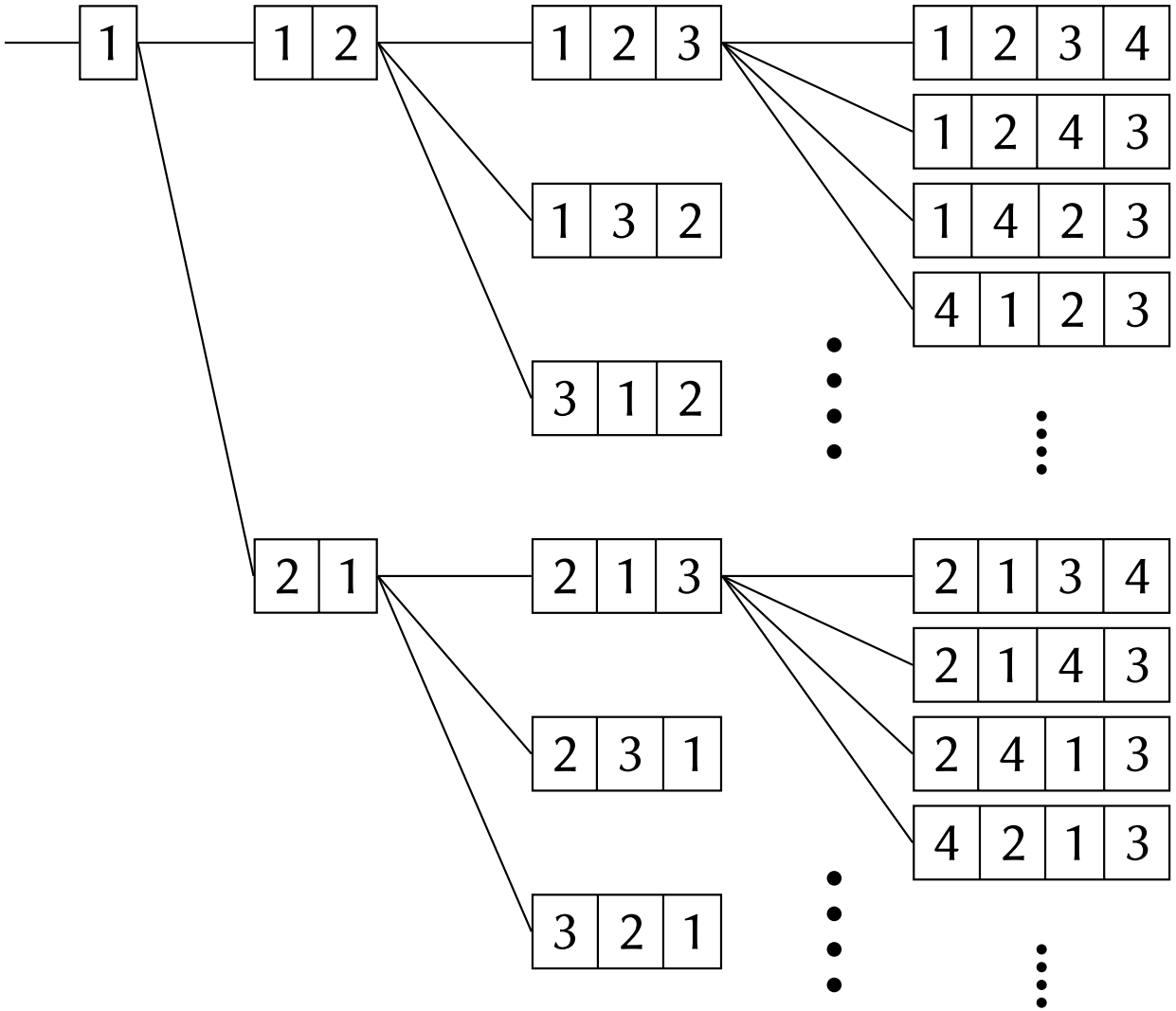
Unterscheidbare Kisten nebeneinander



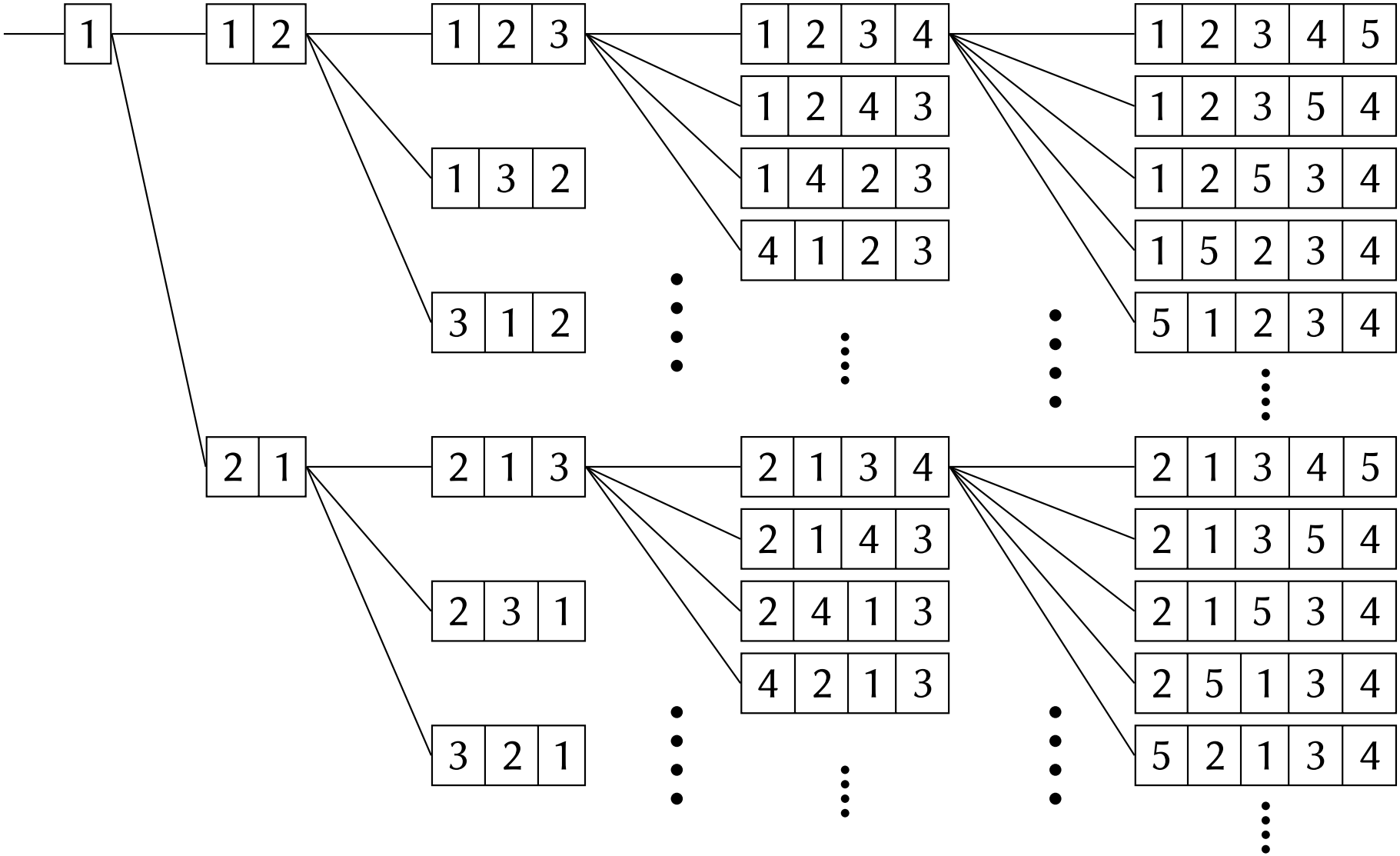
Unterscheidbare Kisten nebeneinander



Unterscheidbare Kisten nebeneinander



Unterscheidbare Kisten nebeneinander



Die Fakultätsfunktion: $n! = 1 \cdot 2 \cdots n$

Die Fakultätsfunktion: $n! = 1 \cdot 2 \cdots n$

Definition ohne Punkte

$$0! = 1$$

$$\text{für jedes } n \in \mathbb{N}_0\text{: } (n+1)! = n! \cdot (n+1)$$

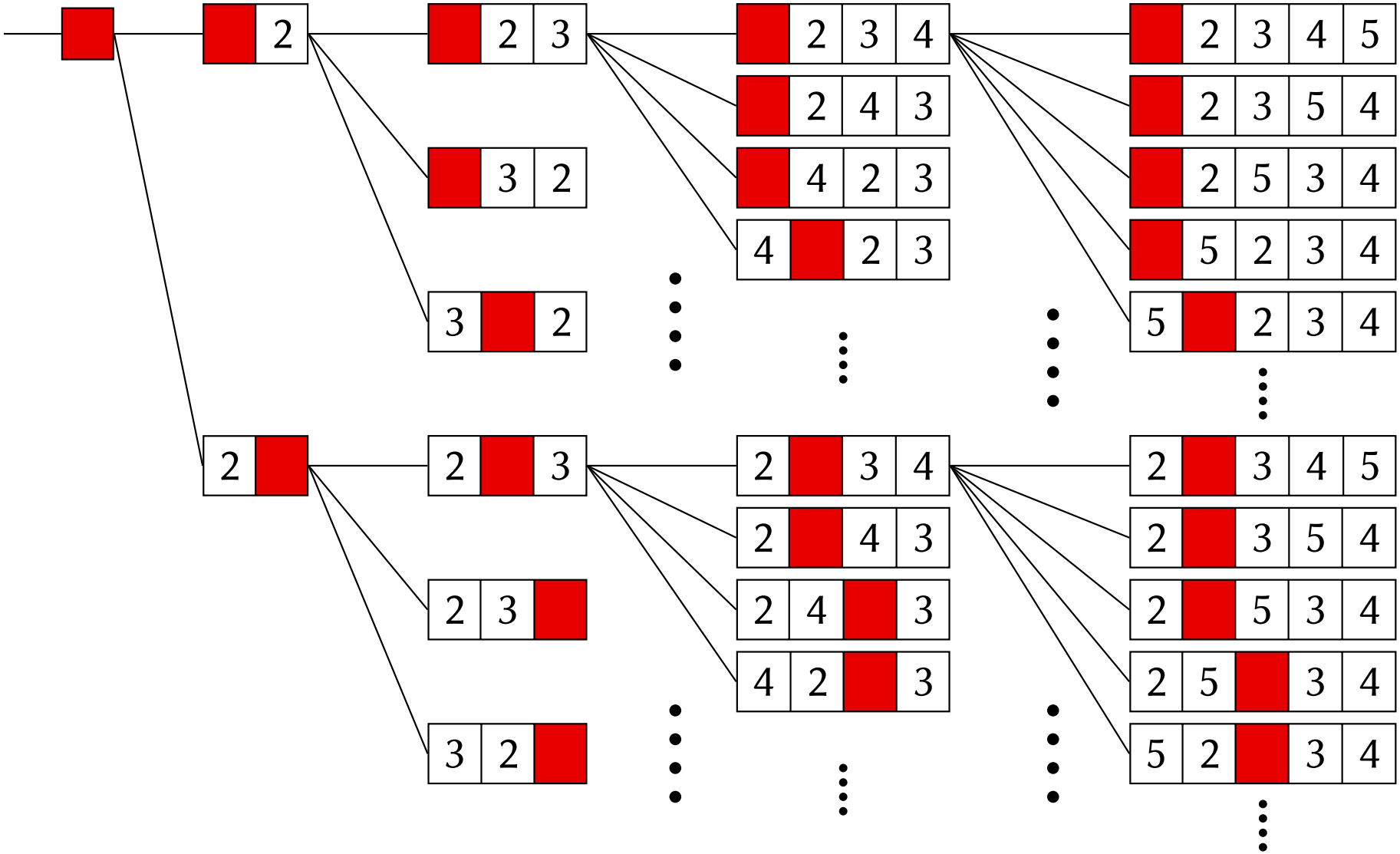
Beispielrechnung:

$$\begin{aligned} 4! &= 3! \cdot 4 \\ &= 2! \cdot 3 \cdot 4 \\ &= 1! \cdot 2 \cdot 3 \cdot 4 \\ &= 0! \cdot 1 \cdot 2 \cdot 3 \cdot 4 \\ &= 1 \cdot 2 \cdot 3 \cdot 4 \end{aligned}$$

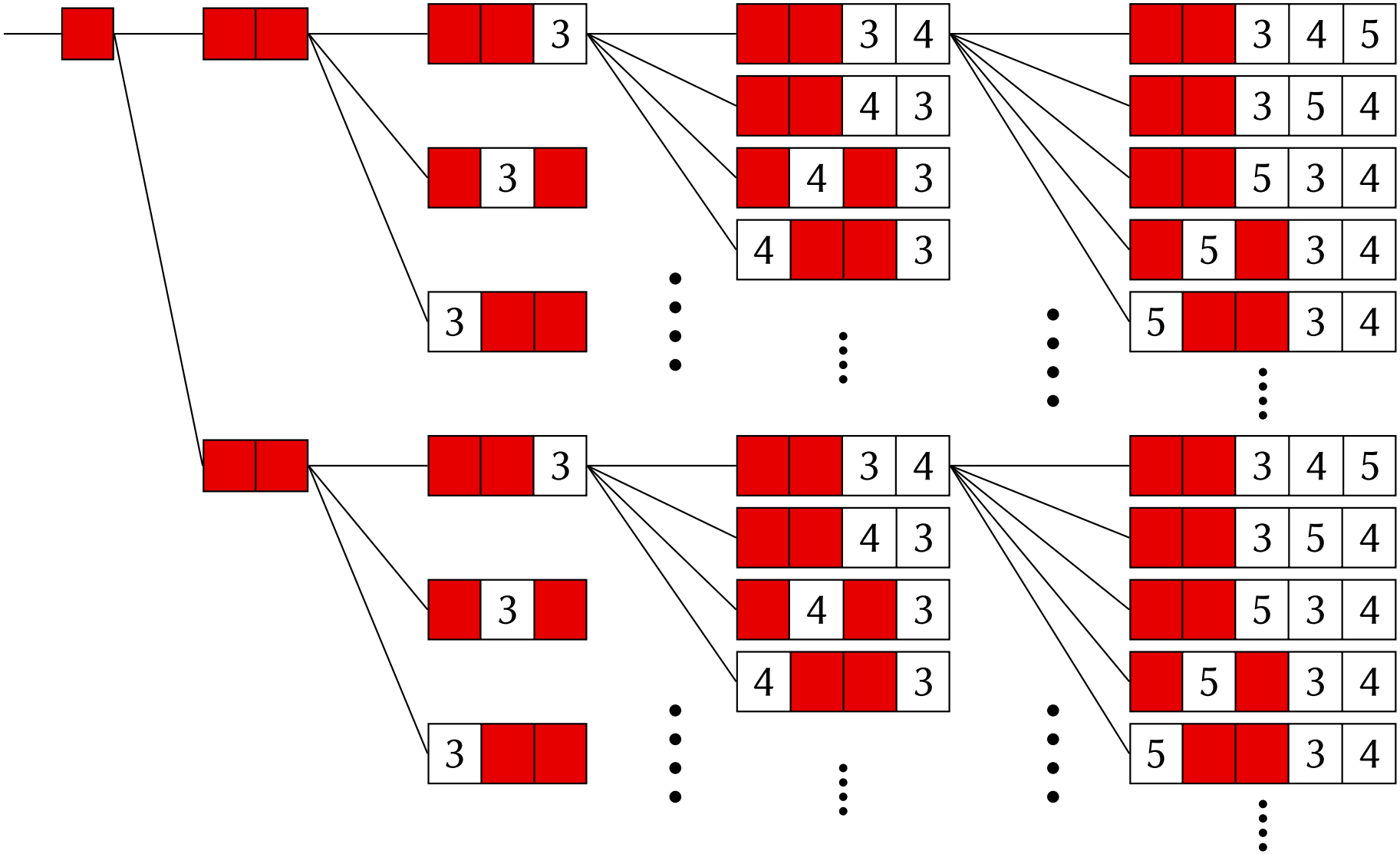
eine Interpretation

- Anzahl verschiedener Reihenfolgen unterscheidbarer Objekte

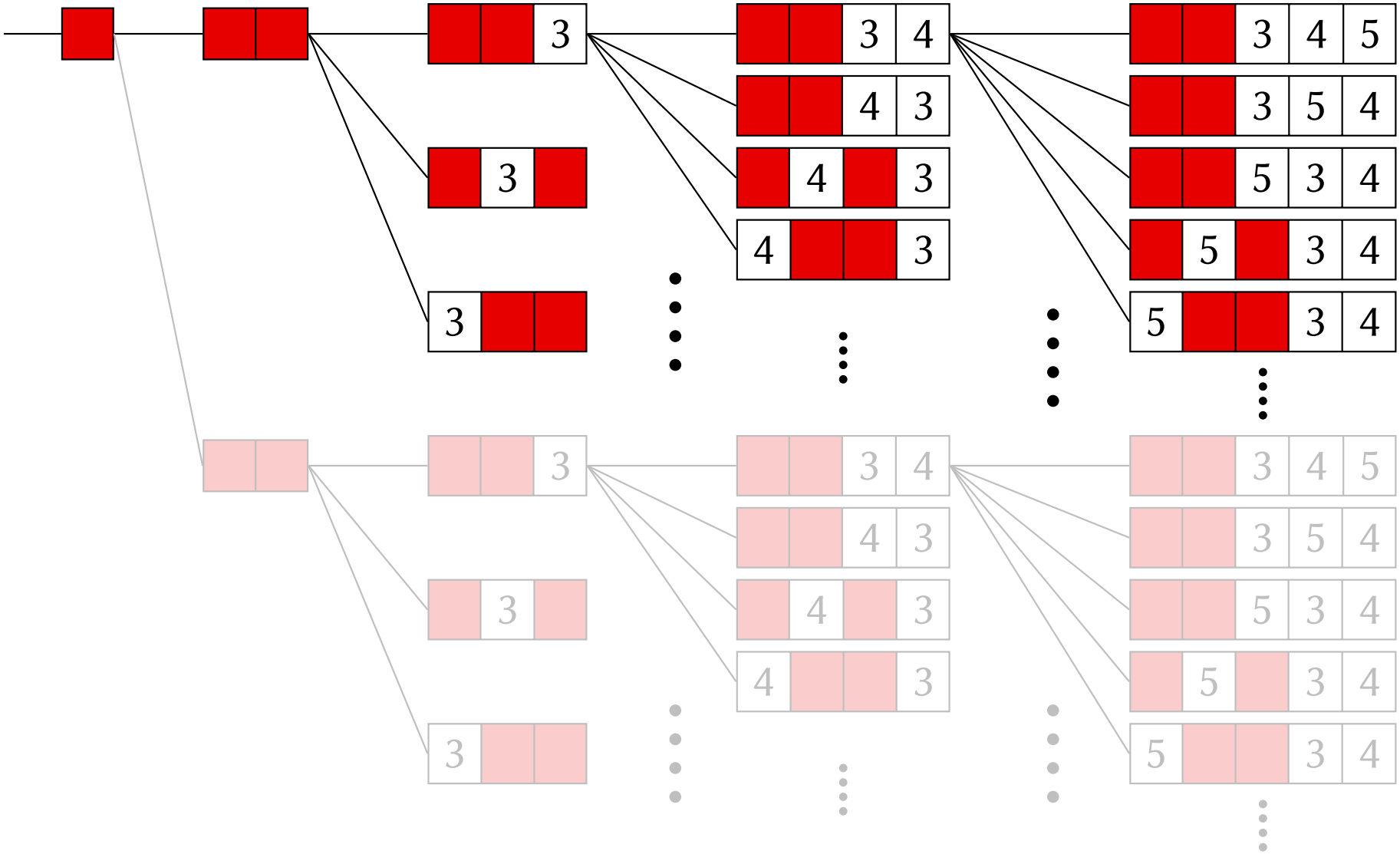
Ununterscheidbare Kisten — erst eine rot



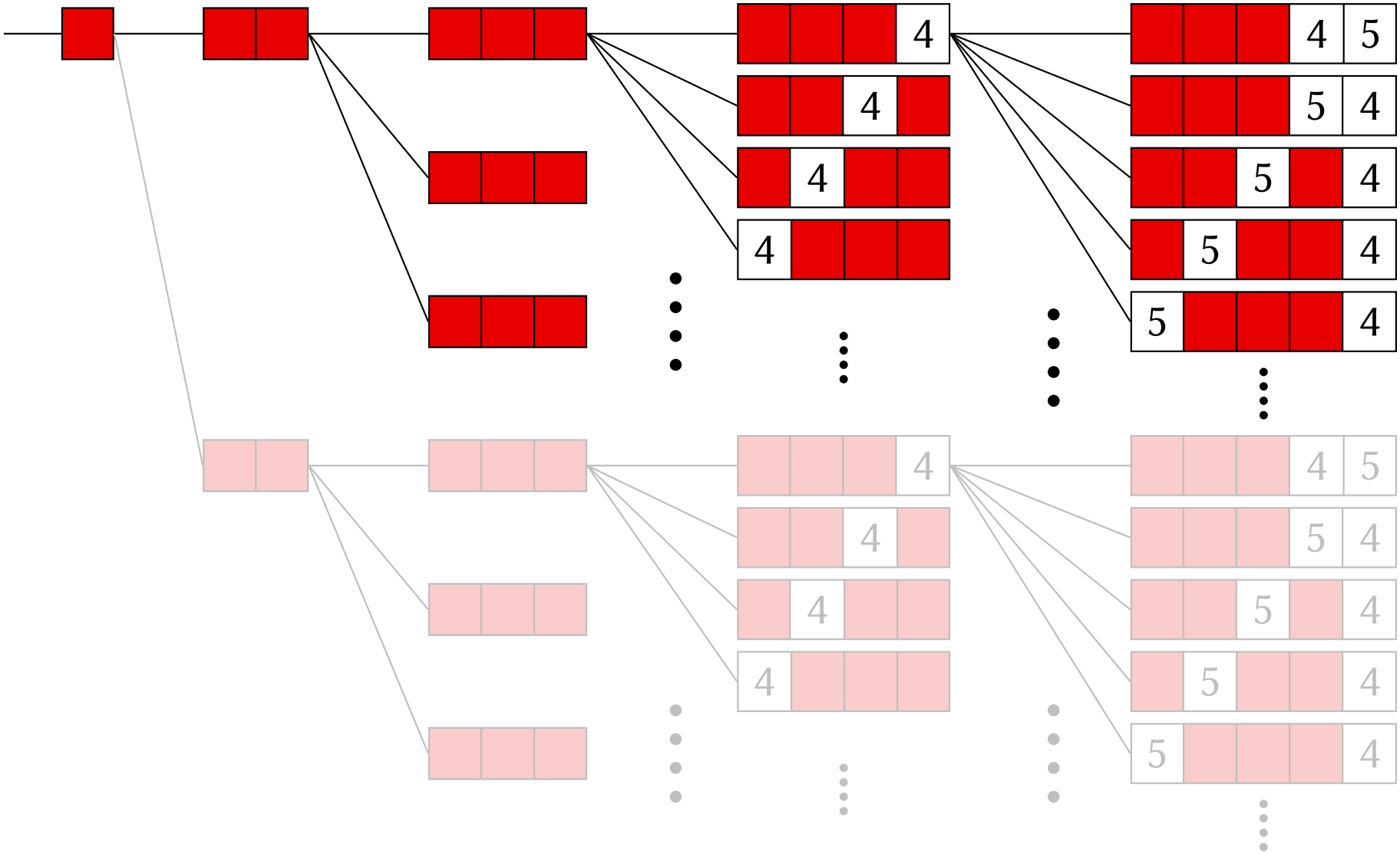
Ununterscheidbare Kisten — dann zwei rot



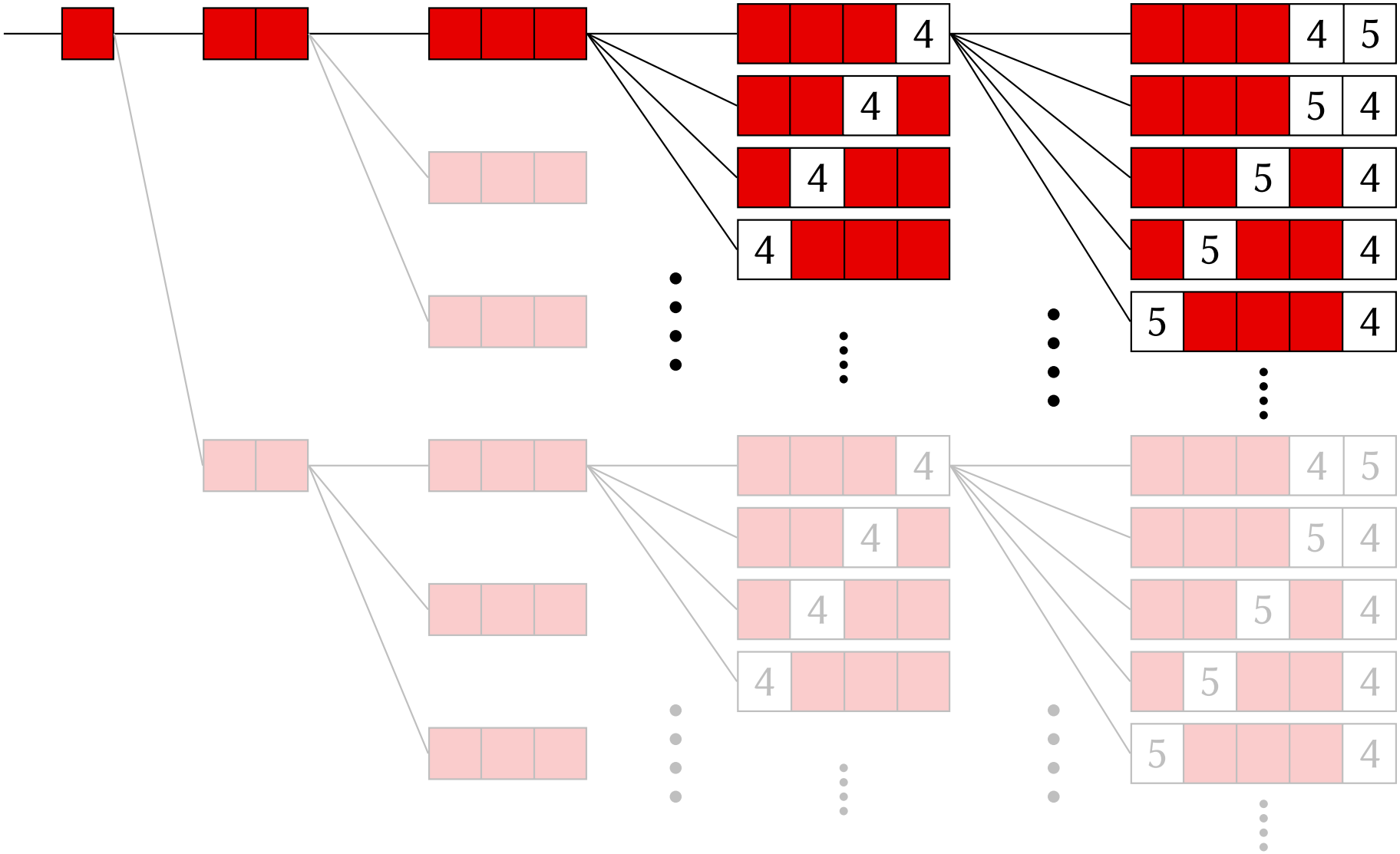
Ununterscheidbare Kisten — dann zwei rot



Ununterscheidbare Kisten — dann drei rot



Ununterscheidbare Kisten — dann drei rot



$n!/k!$ verschiedene Reihenfolgen — wenn k Kisten rot

von den $n!$ Anordnungen jeweils $k!$ ununterscheidbar

also bleiben $\frac{n!}{k!}$

unabhängig davon, welche ununterscheidbar sind

- z. B. auch wenn die größten Zahlen rot gemacht werden

Und nun die restlichen Kisten blau

k rote und $n - k$ blaue Kisten

von den $n!/k!$ Anordnungen jeweils $(n - k)!$ ununterscheidbar

also bleiben $\dfrac{n!}{k!(n-k)!}$

für ganze Zahlen $n \geq 0$ und $0 \leq k \leq n$ nennt man

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

einen Binomialkoeffizienten

- gelesen „n über k“ (oder „k aus n“, *engl.* „n choose k“)

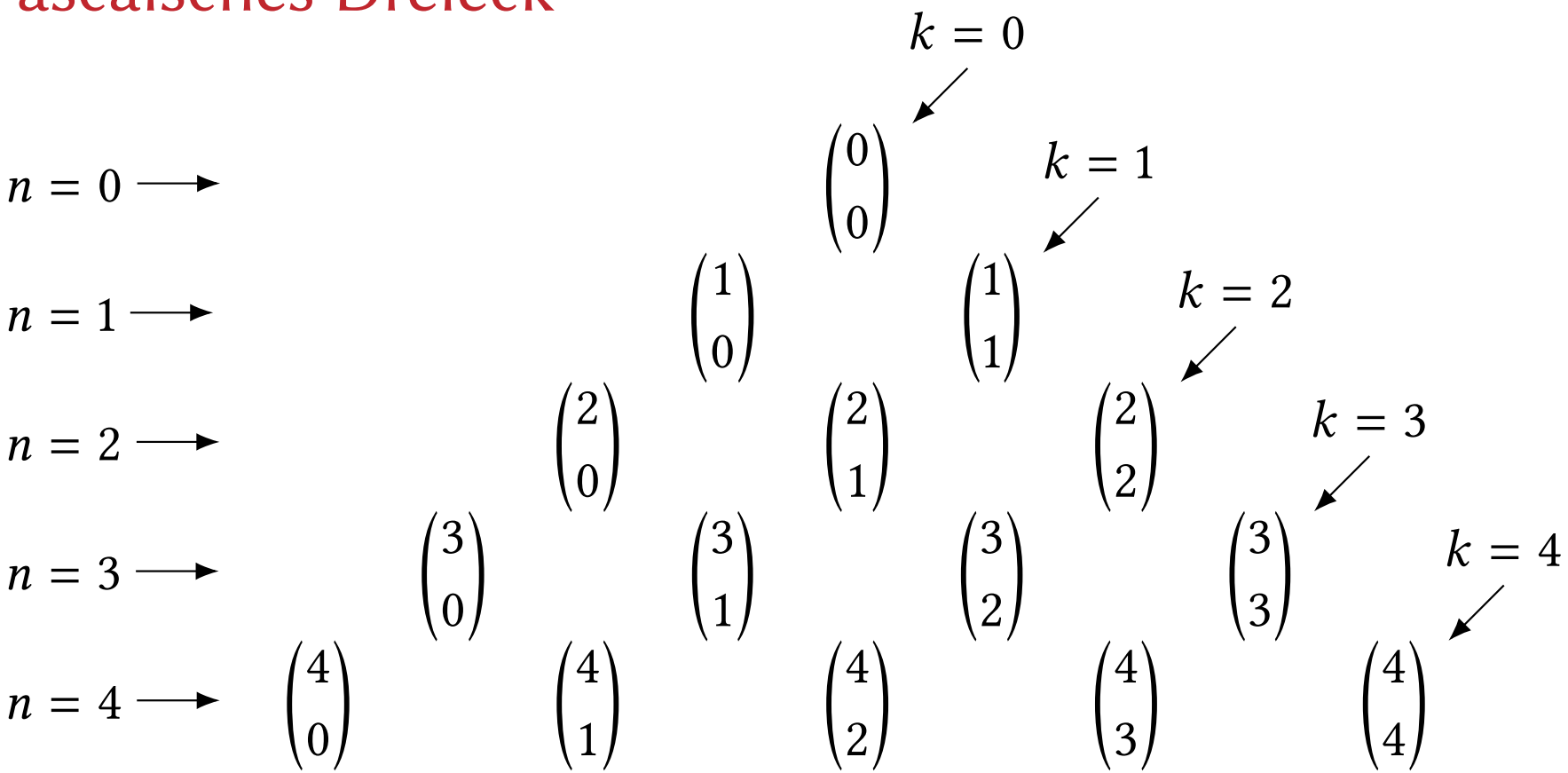
Binomialkoeffizienten

$$\frac{n!}{k!(n-k)!} = \frac{n \cdot (n-1) \cdots (n-k+1) \;\cdot\; (n-k) \cdots 2 \cdot 1}{1 \cdot 2 \cdots k \qquad\quad \cdot\; (n-k) \cdots 2 \cdot 1}$$
$$= \frac{n \cdot (n-1) \cdots (n-k+1)}{1 \cdot 2 \cdots k}$$

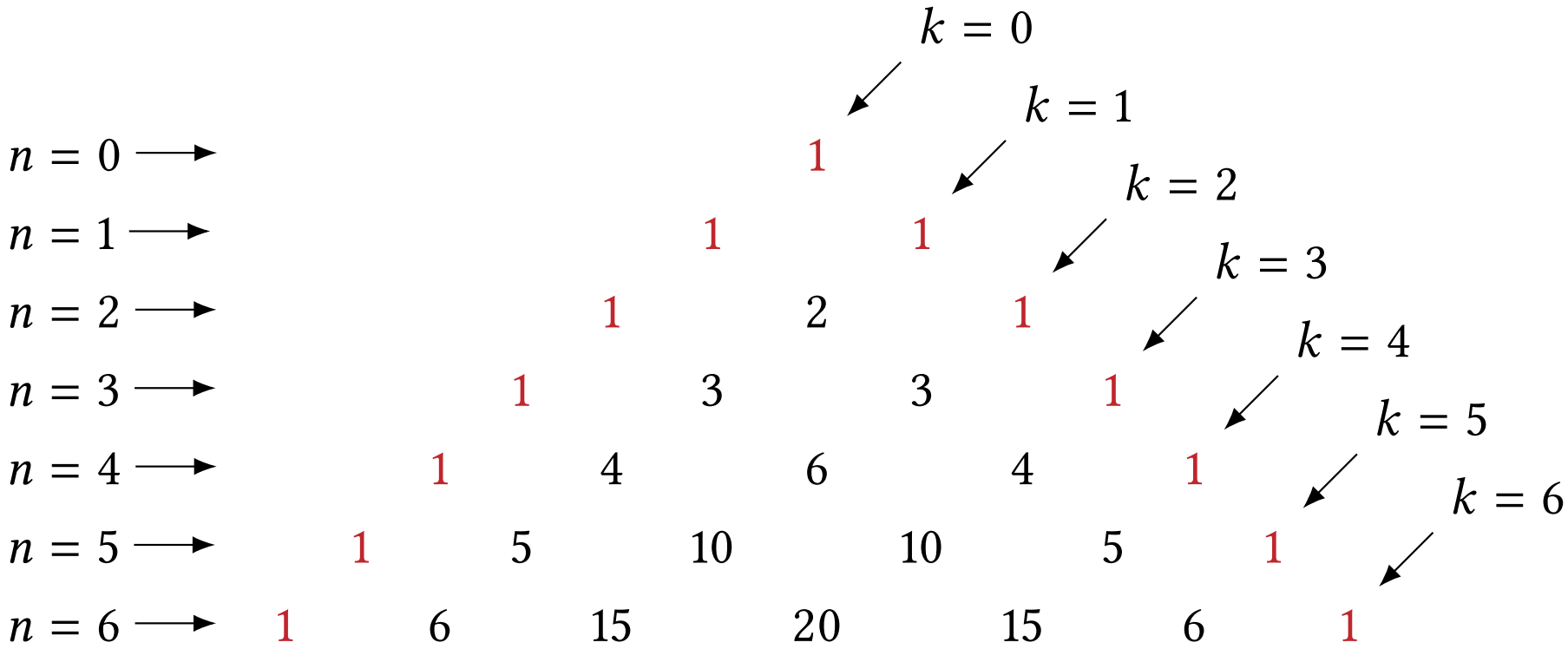
- in Zähler und Nenner gleich viele Faktoren

übrigens: $\binom{n}{0} = \binom{n}{n} = 1$

Pascalsches Dreieck



Pascalsches Dreieck — konkrete Zahlen



Buchstaben statt Farben

Alphabet A mit Symbolen `a` und `b` (und …)

- statt Farben

Wörter $w \in A^*$

- statt aneinandergereihter bunter Kisten

Anzahl Wörter der Länge n

- insgesamt: 2^n
- mit einem `a` an genau k Stellen ($0 \leq k \leq n$):

Buchstaben statt Farben

Alphabet A mit Symbolen a und b (und …)

- statt Farben

Wörter $w \in A^*$

- statt aneinandergereihter bunter Kisten

Anzahl Wörter der Länge n

- insgesamt: 2^n
- mit einem a an genau k Stellen ($0 \leq k \leq n$): $\binom{n}{k}$
- also auch

$$\sum_{i=0}^{n} \binom{n}{i} = \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = 2^n$$

Rückwärts hangeln

es sei $w \in A^*$ ein Wort

- der Länge $n \geq 2$ und
- mit a an genau k Stellen, $1 \leq k \leq n-1$

das letzte Symbol von w?

Rückwärts hangeln

es sei $w \in A^*$ ein Wort

- der Länge $n \geq 2$ und
- mit `a` an genau k Stellen, $1 \leq k \leq n-1$

das letzte Symbol von w?

- ein `a` oder
- ein `b`

davor ein Wort $v \in A^*$ der Länge $n-1$, also

- $w = v\texttt{a}$
- $w = v\texttt{b}$

Rückwärts hangeln

es sei $w \in A^*$ ein Wort

- der Länge $n \geq 2$ und
- mit `a` an genau k Stellen, $1 \leq k \leq n-1$

das letzte Symbol von w?

- ein `a` oder
- ein `b`

davor ein Wort $v \in A^*$ der Länge $n-1$, also

- $w = v$`a` und v enthält an $k-1$ Stellen `a`
- $w = v$`b`

Rückwärts hangeln

es sei $w \in A^*$ ein Wort

- der Länge $n \geq 2$ und
- mit `a` an genau k Stellen, $1 \leq k \leq n-1$

das letzte Symbol von w?

- ein `a` oder
- ein `b`

davor ein Wort $v \in A^*$ der Länge $n-1$, also

- $w = v$`a` und v enthält an $k-1$ Stellen `a`
- $w = v$`b` und v enthält an k Stellen `a`

Rückwärts hangeln

es sei $w \in A^*$ ein Wort

- der Länge $n \geq 2$ und
- mit `a` an genau k Stellen, $1 \leq k \leq n-1$

das letzte Symbol von w?

- ein `a` oder
- ein `b`

davor ein Wort $v \in A^*$ der Länge $n-1$, also

- $w = v$`a` und v enthält an $k-1$ Stellen `a`
- $w = v$`b` und v enthält an k Stellen `a`

also für $n \geq 2$ und $1 \leq k \leq n-1$

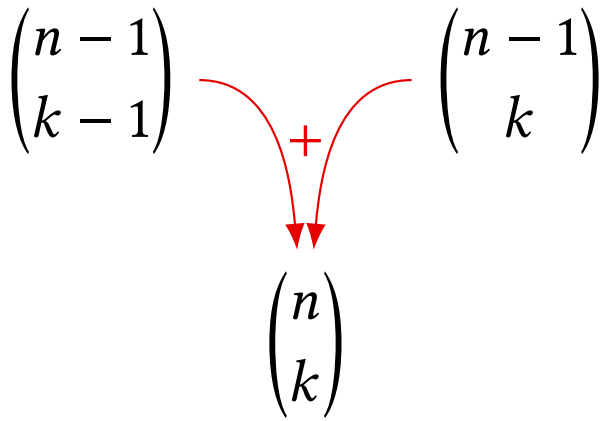
$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Pascalsches Dreieck

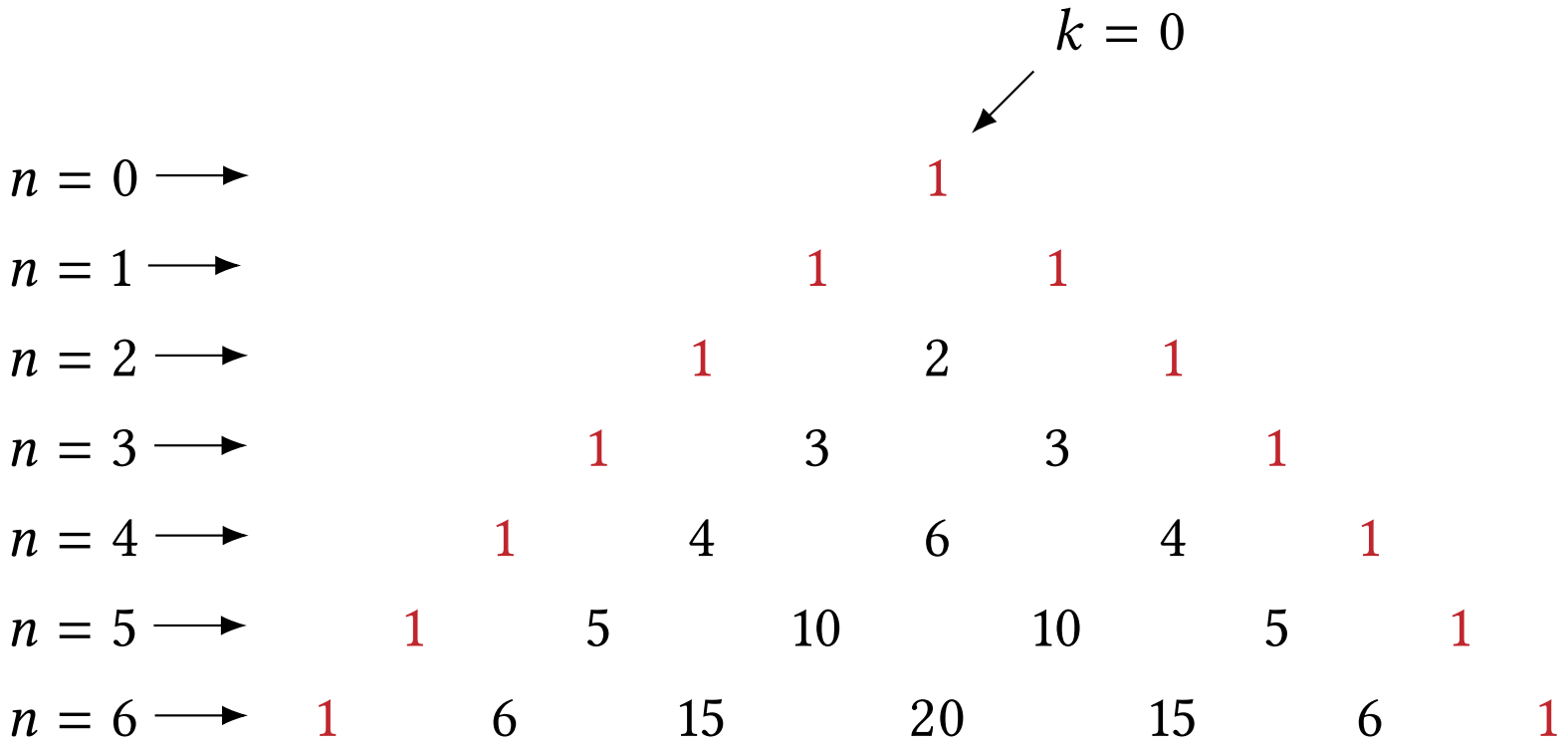
Berechnung von Einträgen

an den Rändern alles 1

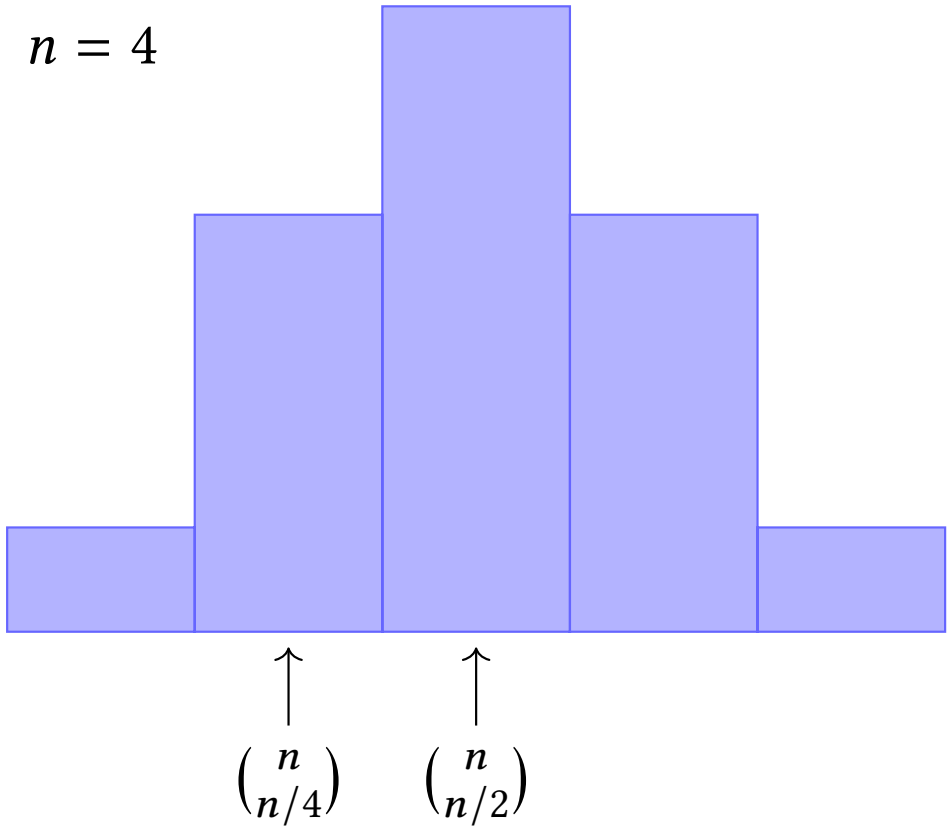
im Innern



Pascalsches Dreieck — konkrete Zahlen

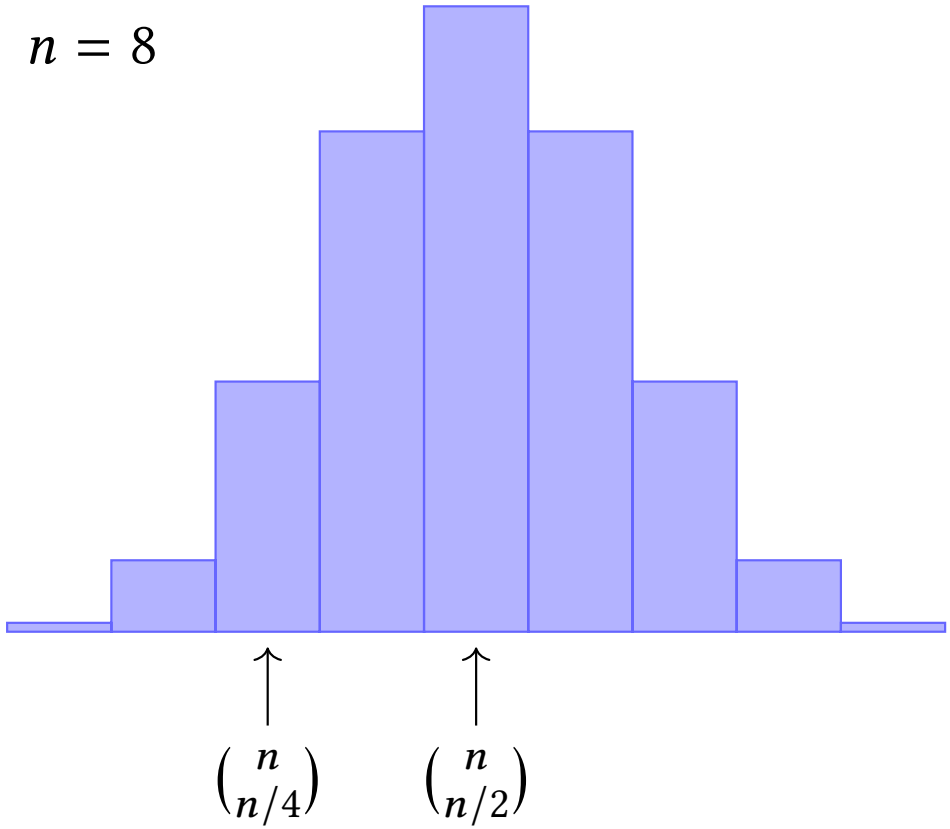


Binomialkoeffizienten — nummerische Verteilung



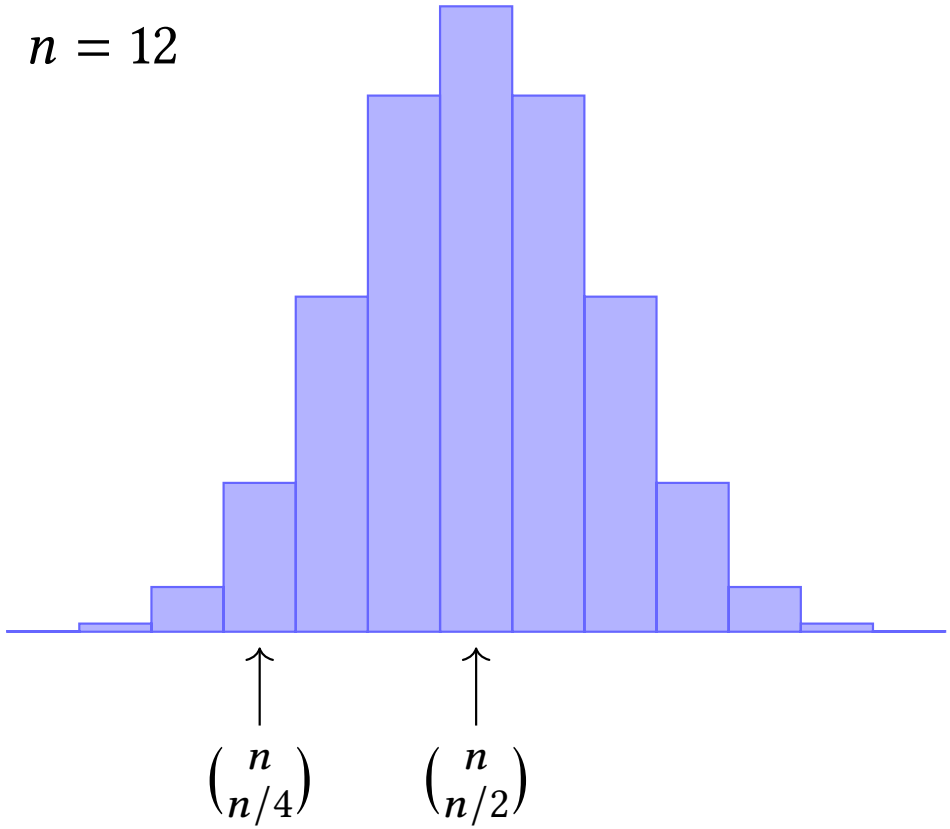
normiert auf gleiche maximale Höhe und gleiche Breite

Binomialkoeffizienten — nummerische Verteilung



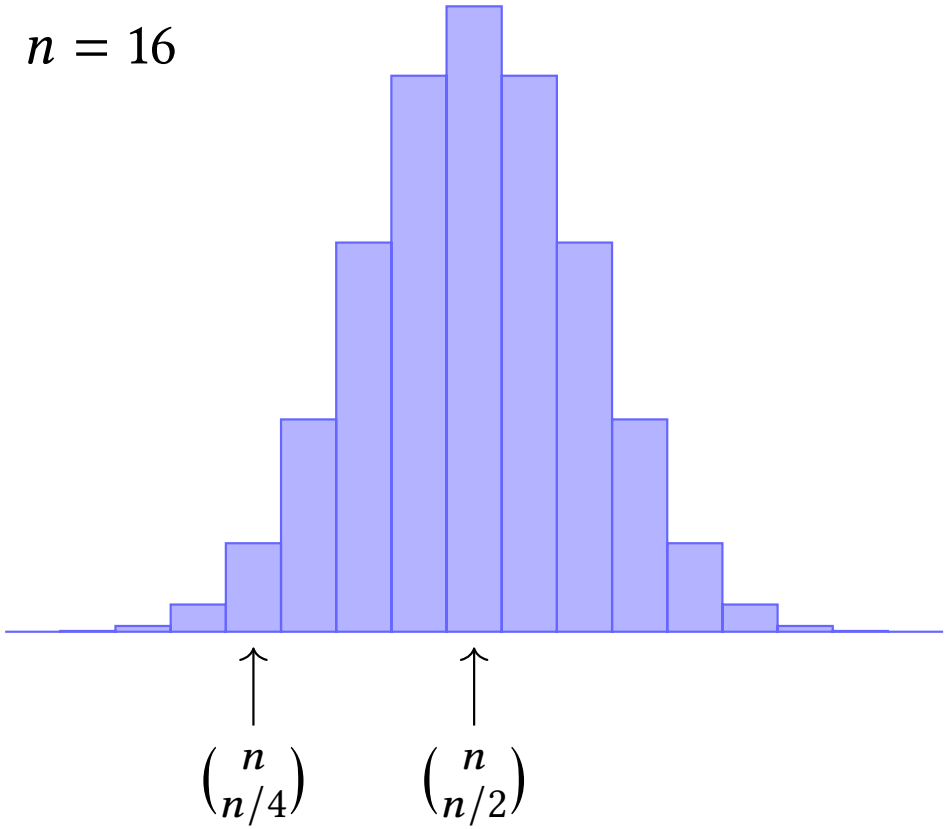
normiert auf gleiche maximale Höhe und gleiche Breite

Binomialkoeffizienten — nummerische Verteilung



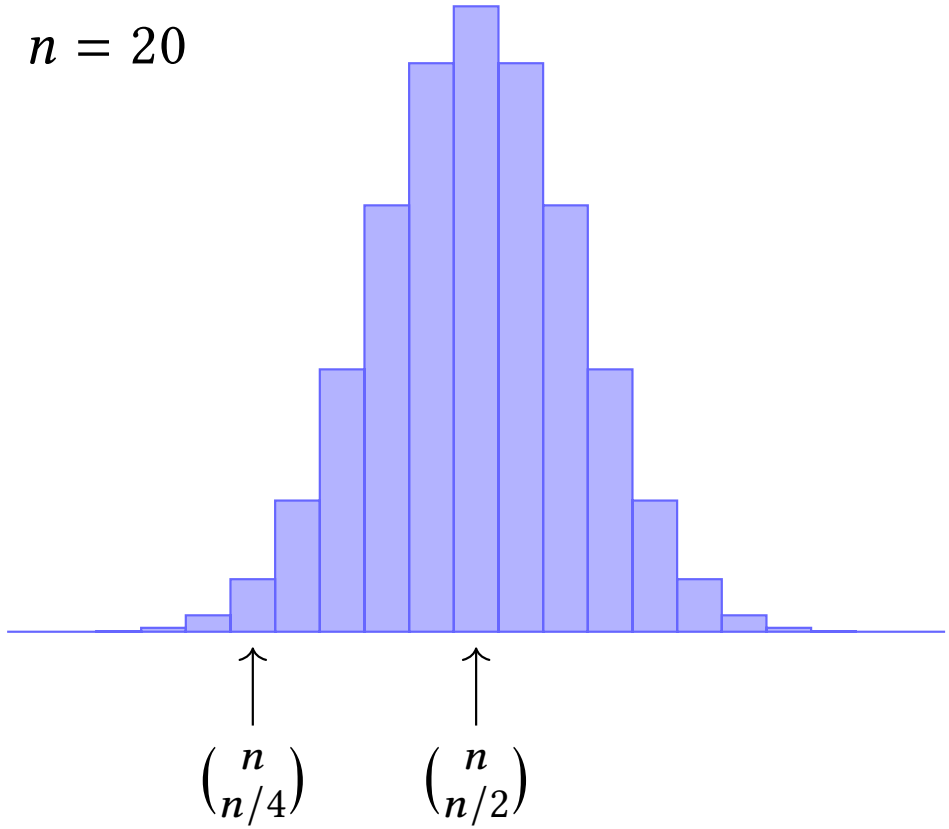
normiert auf gleiche maximale Höhe und gleiche Breite

Binomialkoeffizienten — nummerische Verteilung



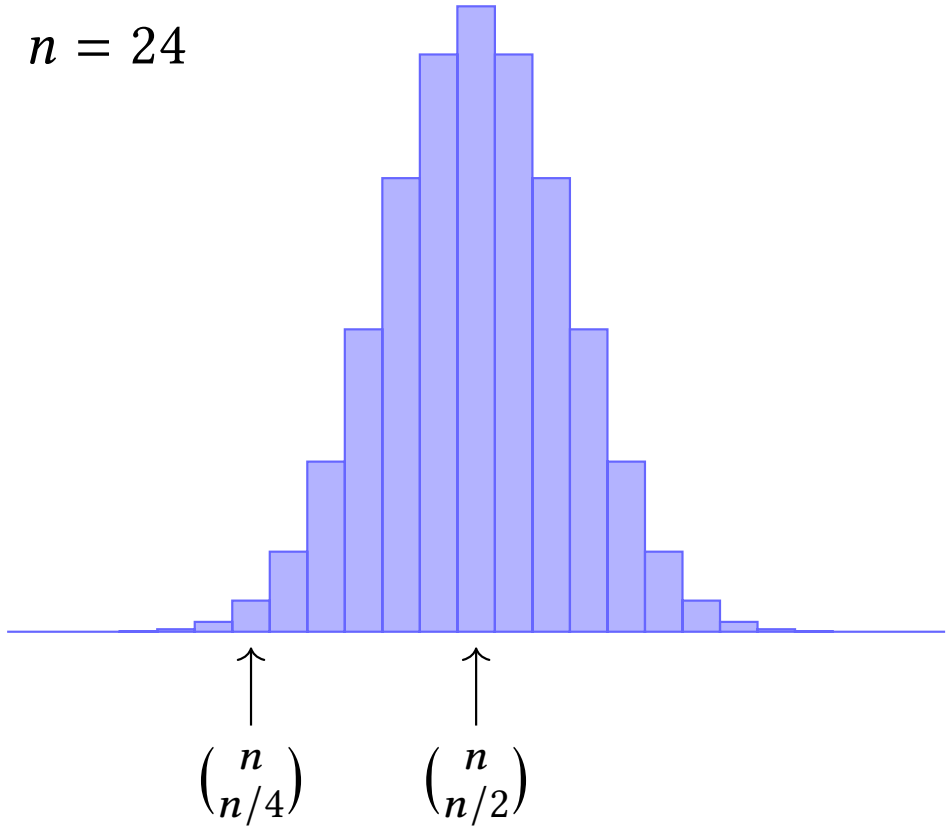
normiert auf gleiche maximale Höhe und gleiche Breite

Binomialkoeffizienten — nummerische Verteilung



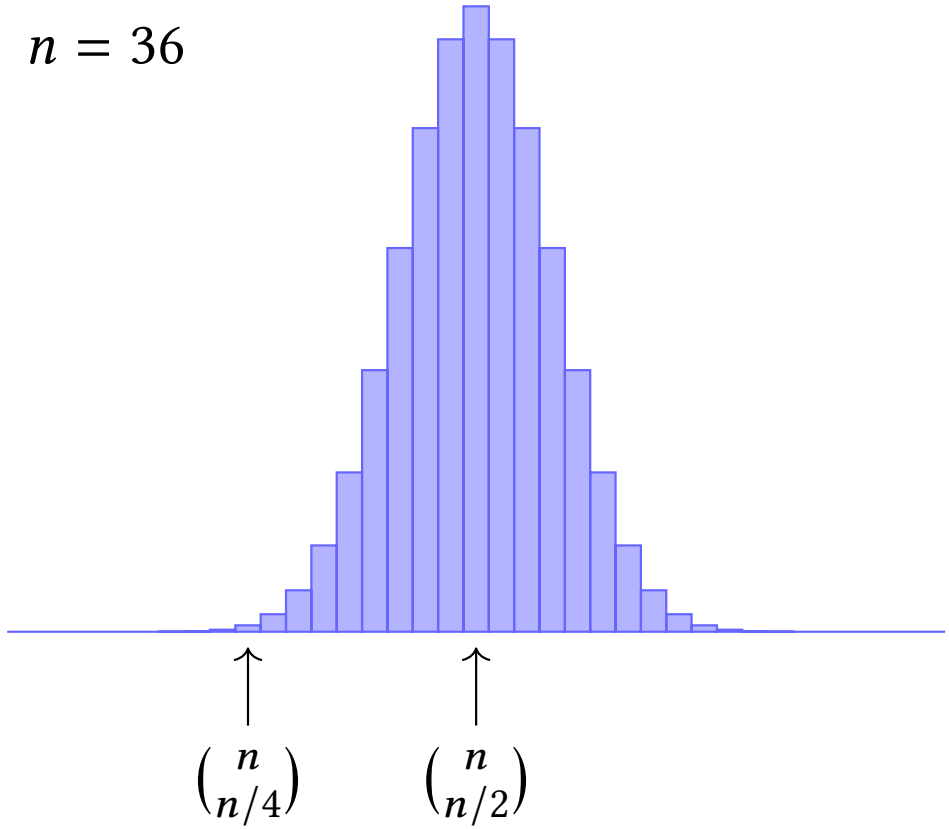
normiert auf gleiche maximale Höhe und gleiche Breite

Binomialkoeffizienten — nummerische Verteilung



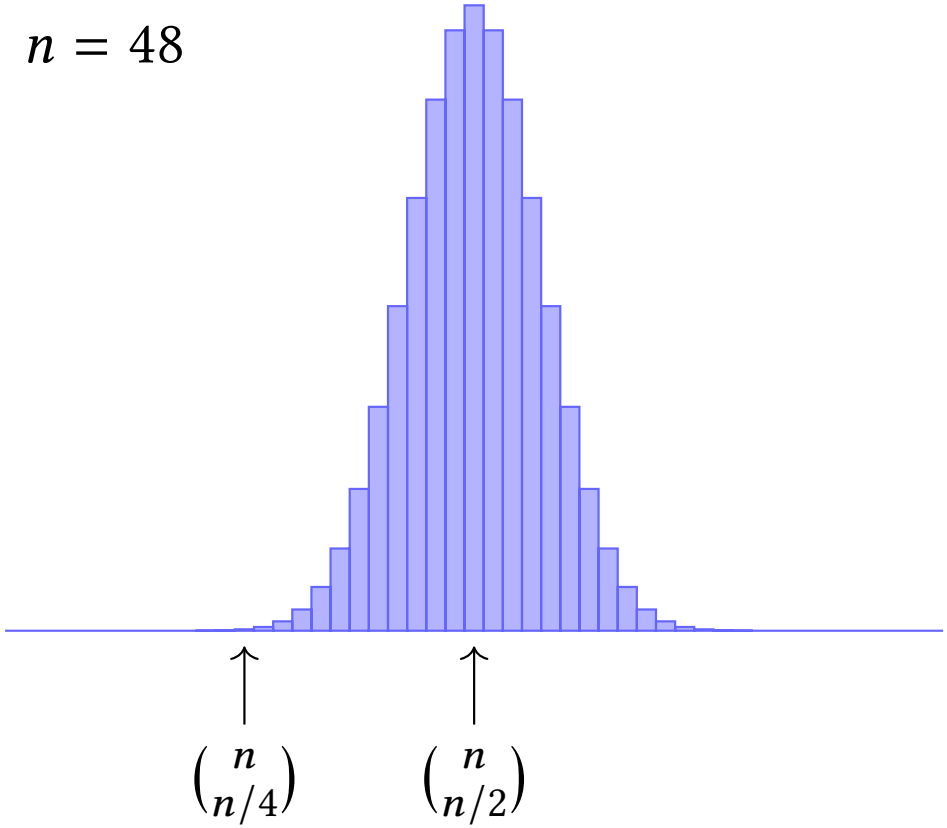
normiert auf gleiche maximale Höhe und gleiche Breite

Binomialkoeffizienten — nummerische Verteilung



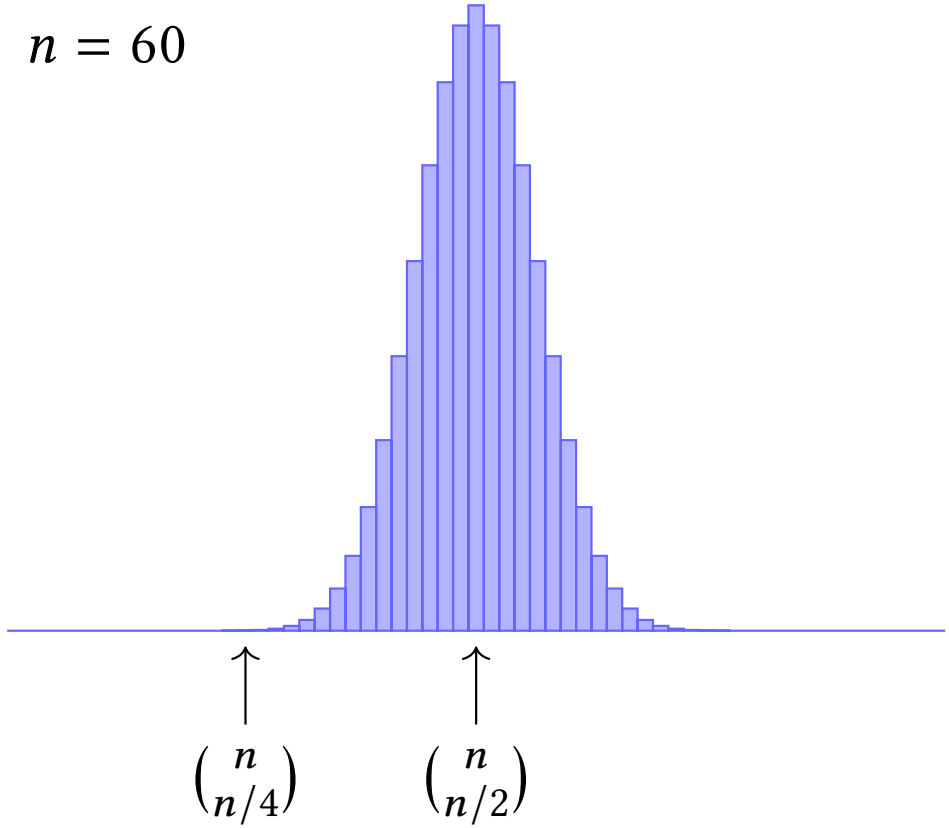
normiert auf gleiche maximale Höhe und gleiche Breite

Binomialkoeffizienten — nummerische Verteilung



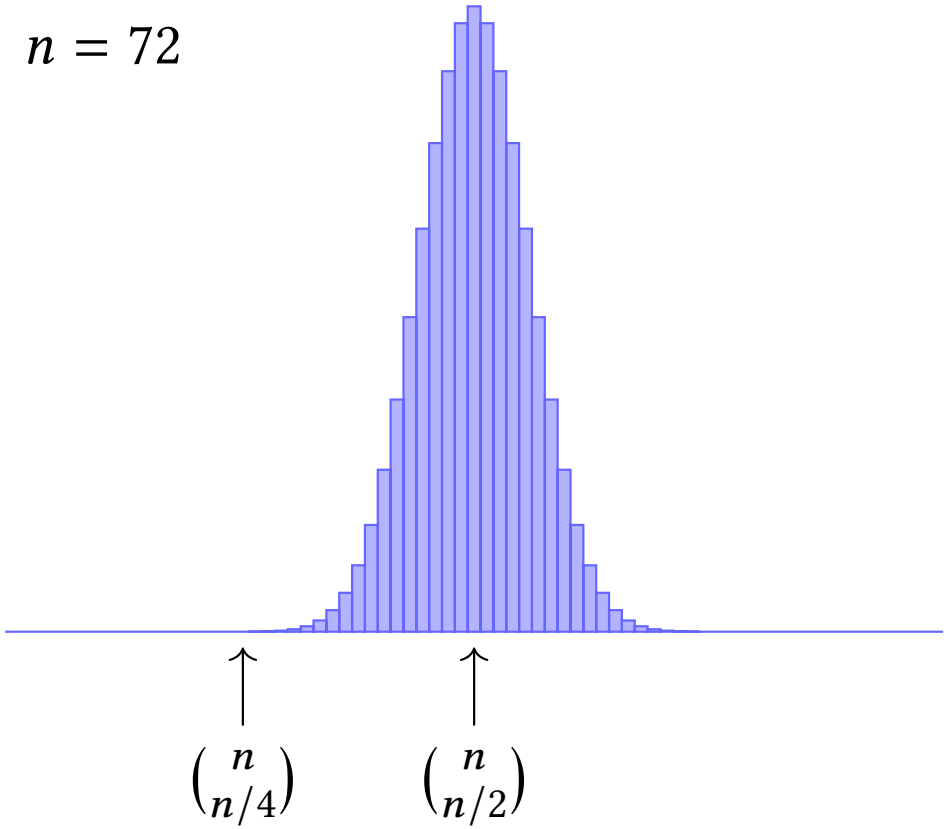
normiert auf gleiche maximale Höhe und gleiche Breite

Binomialkoeffizienten — nummerische Verteilung



normiert auf gleiche maximale Höhe und gleiche Breite

Binomialkoeffizienten — nummerische Verteilung



normiert auf gleiche maximale Höhe und gleiche Breite

Binomialkoeffizienten — zwei Näherungen

$$\binom{n}{n/2} \approx \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{n}} \cdot \frac{2}{\sqrt{1}} \cdot 2^n \qquad = \frac{2}{\sqrt{2\pi n}} \cdot 2^n$$

$$\binom{n}{n/4} \approx \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{n}} \cdot \frac{4}{\sqrt{3}} \cdot \left(\frac{4}{3^{3/4}}\right)^n \approx \frac{2.309}{\sqrt{2\pi n}} \cdot 1.755^n$$

Ein Ausschnitt aus *Grundbegriffe der Informatik*

Huffman-Codierung

Huffman-Codierung — ein Überblick

gegeben: Alphabet A und Wort $w \in A^*$

Huffman-Codierung

- berechnet mit Hilfe von w
- bildet jedes Zeichen aus A ab
  auf ein Wort aus 0 und 1
- typischerweise angewendet auf w

im Allgemeinen macht das die Wörter länger

Varianten davon Bestandteil von
Kompressionsverfahren gzip, bzip2

Kompression von (natürlichsprachlichen) Texten

Wie kann man jedes Wort aus $\{\mathtt{a}, \mathtt{b}\}^*$s einer Länge n durch ein kürzeres ersetzen?

Kompression von (natürlichsprachlichen) Texten

Wie kann man jedes Wort aus $\{\mathtt{a}, \mathtt{b}\}^*$s einer Länge n durch ein kürzeres ersetzen?

- radikal: man ersetzt jedes Wort durch ein einziges $\mathtt{a}$
  - keine Dekompression

Kompression von (natürlichsprachlichen) Texten

Wie kann man jedes Wort aus $\{\mathtt{a}, \mathtt{b}\}^*$s einer Länge n durch ein kürzeres ersetzen?

- radikal: man ersetzt jedes Wort durch ein einziges $\mathtt{a}$
  - keine Dekompression
- verlustfrei und echt verkürzend?

$\mathtt{aaa}$	$\mapsto$	?	$\mathtt{baa}$	$\mapsto$	?
$\mathtt{aab}$	$\mapsto$	?	$\mathtt{bab}$	$\mapsto$	?
$\mathtt{aba}$	$\mapsto$	?	$\mathtt{bba}$	$\mapsto$	?
$\mathtt{abb}$	$\mapsto$	?	$\mathtt{bbb}$	$\mapsto$	?

Kompression von (natürlichsprachlichen) Texten

Wie kann man jedes Wort aus $\{\mathtt{a}, \mathtt{b}\}^*$s einer Länge n durch ein kürzeres ersetzen?

- radikal: man ersetzt jedes Wort durch ein einziges $\mathtt{a}$
  - keine Dekompression
- verlustfrei und echt verkürzend?

$\mathtt{aaa}$	$\mapsto$	?	$\mathtt{baa}$	$\mapsto$	?
$\mathtt{aab}$	$\mapsto$	?	$\mathtt{bab}$	$\mapsto$	?
$\mathtt{aba}$	$\mapsto$	?	$\mathtt{bba}$	$\mapsto$	?
$\mathtt{abb}$	$\mapsto$	?	$\mathtt{bbb}$	$\mapsto$	?

- durch Vergrößerung des Zielalphabets
  - „gemogelt“

Kompression von (natürlichsprachlichen) Texten

Wie kann man jedes Wort aus $\{\mathtt{a}, \mathtt{b}\}^*$s einer Länge n durch ein kürzeres ersetzen?

- radikal: man ersetzt jedes Wort durch ein einziges $\mathtt{a}$
  - keine Dekompression
- verlustfrei und echt verkürzend?

$\mathtt{aaa}$	$\mapsto$	?	$\mathtt{baa}$	$\mapsto$	?
$\mathtt{aab}$	$\mapsto$	?	$\mathtt{bab}$	$\mapsto$	?
$\mathtt{aba}$	$\mapsto$	?	$\mathtt{bba}$	$\mapsto$	?
$\mathtt{abb}$	$\mapsto$	?	$\mathtt{bbb}$	$\mapsto$	?

- durch Vergrößerung des Zielalphabets
  - „gemogelt“
- bei gleich großem Zielalphabet
  nicht „für alle Zeichenfolgen gleichzeitig“ möglich

Kompression von Texten (2)

„viel“ weniger sinnvolle Texte als sinnlose

verschiedene Erklärungen, unter anderem:

Kompression von Texten (2)

„viel“ weniger sinnvolle Texte als sinnlose

verschiedene Erklärungen, unter anderem:

stark unterschiedliche Buchstabenhäufigkeiten (in %)

	e	n	i	s	r	a
deutsch	17.40	9.78	7.55	7.27	7.00	6.51
englisch	12.70	6.75	6.97	6.33	5.99	8.17
italienisch	11.79	6.88	11.28	4.98	6.37	11.74

Kompression von Texten (3)

„deutlich“ weniger Texte mit unterschiedlichen Buchstabenhäufigkeiten

zum Beispiel

- gleich viele a und b: $\binom{n}{n/2} \approx \frac{2}{\sqrt{2\pi n}} \cdot 2^n$
- nur ein Viertel a: $\binom{n}{n/4} \approx \frac{2.309}{\sqrt{2\pi n}} \cdot 1.755^n$
- konstant 3 a: $\binom{n}{3} = \frac{n \cdot (n-1) \cdot (n-2)}{6} \approx \frac{n^3}{6} \pm \cdots$

Codierung — einleitende Überlegungen

gegeben 4 Symbole a, b, c, d

- Codierung durch Wörter aus 0 und 1 ?

  a ↦ ?
  b ↦ ?
  c ↦ ?
  d ↦ ?

Codierung — einleitende Überlegungen

gegeben 4 Symbole a, b, c, d

- Codierung durch Wörter aus 0 und 1 ?

  a ↦ 00
  b ↦ 01
  c ↦ 10
  d ↦ 11

Codierung — einleitende Überlegungen

gegeben 4 Symbole a, b, c, d

- Codierung durch Wörter aus 0 und 1 ?

  a ↦ 00
  b ↦ 01
  c ↦ 10
  d ↦ 11

gegeben 3 Symbole a, b, c

- Codierung durch Wörter aus 0 und 1 ?

Codierung — einleitende Überlegungen

gegeben 4 Symbole a, b, c, d

- Codierung durch Wörter aus 0 und 1 ?

  a ↦ 00
  b ↦ 01
  c ↦ 10
  d ↦ 11

gegeben 3 Symbole a, b, c

- Codierung durch Wörter aus 0 und 1 ?

  a ↦ 00
  b ↦ 01
  c ↦ 10

Codierung — einleitende Überlegungen

gegeben 4 Symbole a, b, c, d

- Codierung durch Wörter aus 0 und 1 ?

  a ↦ 00
  b ↦ 01
  c ↦ 10
  d ↦ 11

gegeben 3 Symbole a, b, c

- Codierung durch Wörter aus 0 und 1 ?

  a ↦ 00
  b ↦ 01
  c ↦ 10
- auch für cccbcacccbccccacccccb?

Codierung — einleitende Überlegungen

gegeben 4 Symbole a, b, c, d

- Codierung durch Wörter aus 0 und 1 ?

  a ↦ 00
  b ↦ 01
  c ↦ 10
  d ↦ 11

gegeben 3 Symbole a, b, c

- Codierung durch Wörter aus 0 und 1 ?

  a ↦ 00
  b ↦ 01
  c ↦ 10

- auch für cccbcacccbccccacccccb?
  - c ↦ 1 tut es auch

Huffman-Codierung — Voraussetzungen und Plan

Gegeben

- Wort w aus Zeichen in A
- $N_x(w)$: wie oft kommt Zeichen x in w vor
- alle $N_x(w) > 0$

Bestimmung eines Huffman-Codes in zwei Phasen

1. Konstruktion eines „Baumes“
   - Blätter entsprechen den $x \in A$ und
   - Kanten mit 0 und 1 beschriftet
2. Ablesen der Codes aus dem Baum

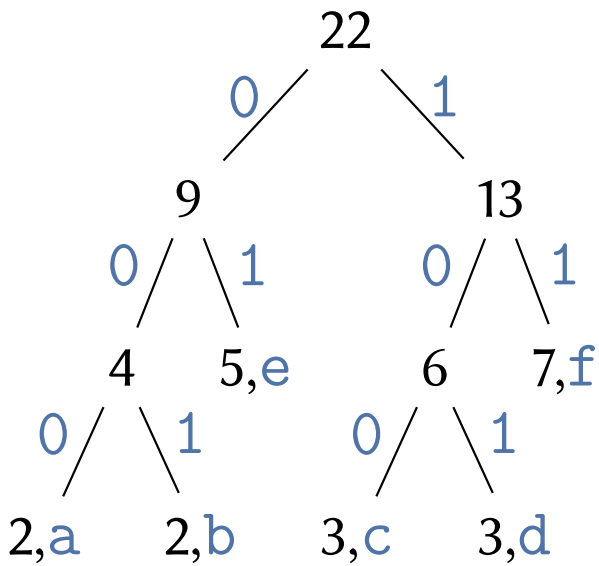
Huffman-Codierung — Beispiel Ein- und Ausgabe

Eingabe: $w =$ `afebfecaffdeddccefbeff`

Huffman-Codierung — Beispiel Ein- und Ausgabe

Eingabe: $w =$ `afebfecaffdeddccefbeff`

Baum am Ende:



Codierung:

x	a	b	c	d	e	f
h(x)	000	001	100	101	01	11

Konstruktion des Huffman-Baumes (1)

Anfang: jedes Zeichen mit seiner Häufigkeit

5,e 7,f

2,a 2,b 3,c 3,d

nennen wir sie „*freie*“ Knoten

Konstruktion des Huffman-Baumes (1)

Anfang: jedes Zeichen mit seiner Häufigkeit

5,e 7,f

2,a 2,b 3,c 3,d

nennen wir sie „*freie*“ Knoten

danach: solange noch mindestens zwei freie Knoten

- zwei freie Knoten mit kleinsten Häufigkeiten k_1, k_2
- „zusammenführen“ zu neuem freien Knoten mit $k_1 + k_2$ und
- Kanten von ehemals freien Knoten zum neuen

Konstruktion des Huffman-Baumes (2)

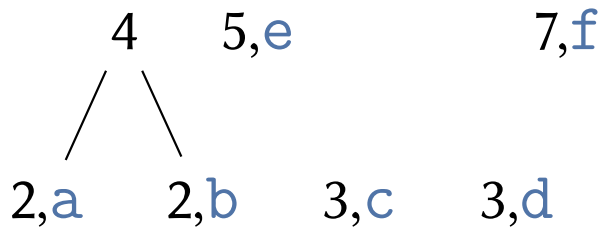
Beispiel

5,e 7,f

2,a 2,b 3,c 3,d

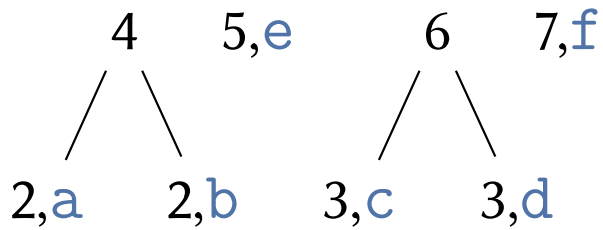
Konstruktion des Huffman-Baumes (2)

Beispiel



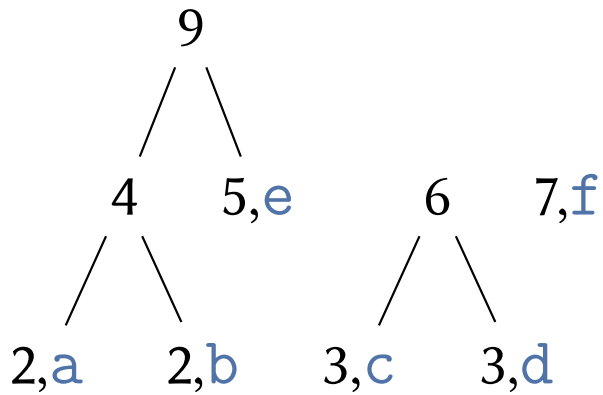
Konstruktion des Huffman-Baumes (3)

Beispiel



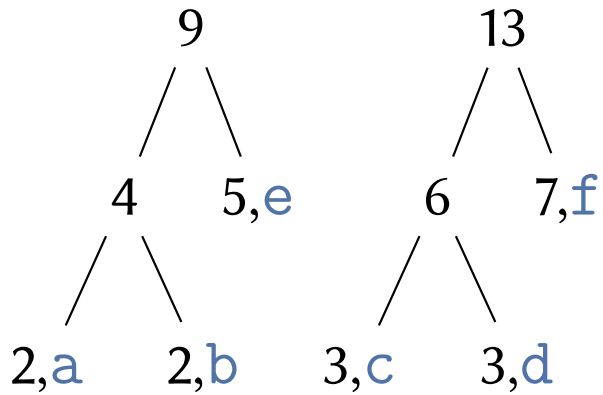
Konstruktion des Huffman-Baumes (4)

Beispiel



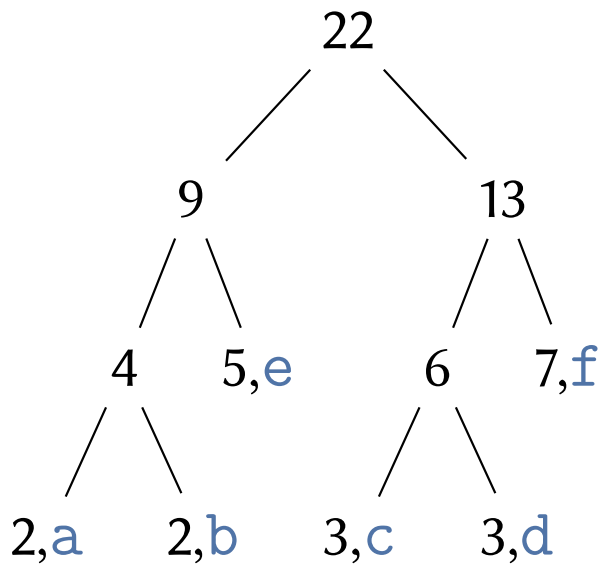
Konstruktion des Huffman-Baumes (5)

Beispiel

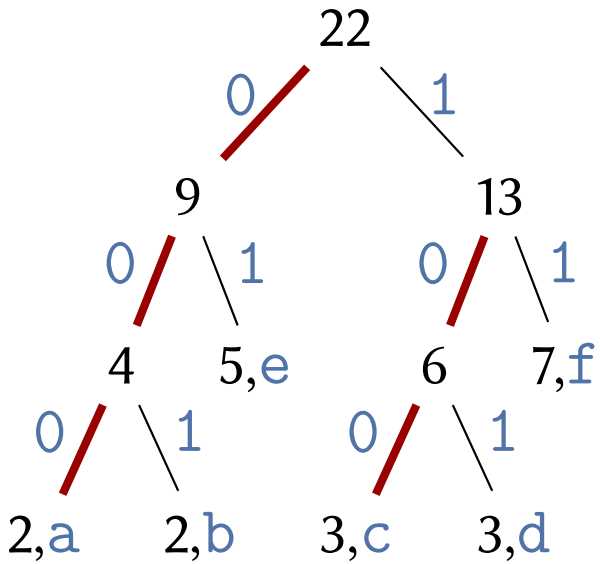


Konstruktion des Huffman-Baumes (6)

Beispiel



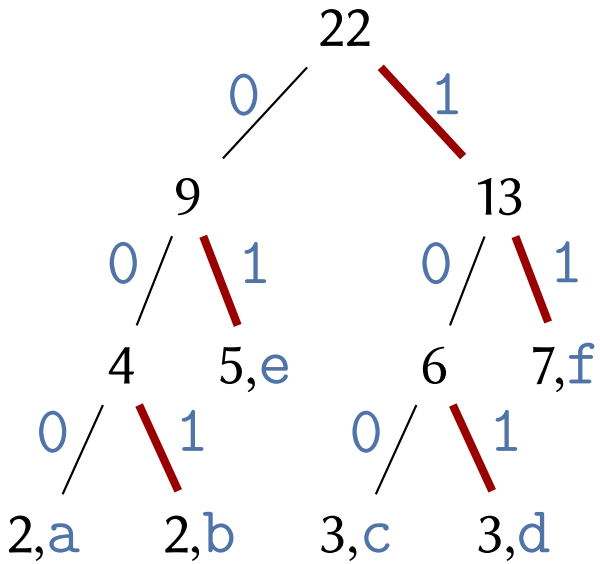
Beschriftung der Kanten



nach links führende Kanten
mit 0 beschriftet

nach rechts führende Kanten
mit 1 beschriftet

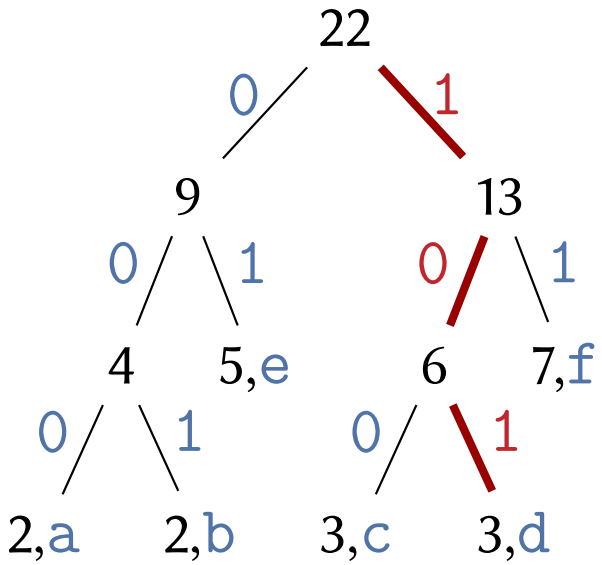
Beschriftung der Kanten



nach links führende Kanten
mit 0 beschriftet

nach rechts führende Kanten
mit 1 beschriftet

Ablesen der Codierungen



gehe auf kürzestem Weg

- von der Wurzel des Baumes
- zu dem Blatt für x

konkateniere der Reihe nach

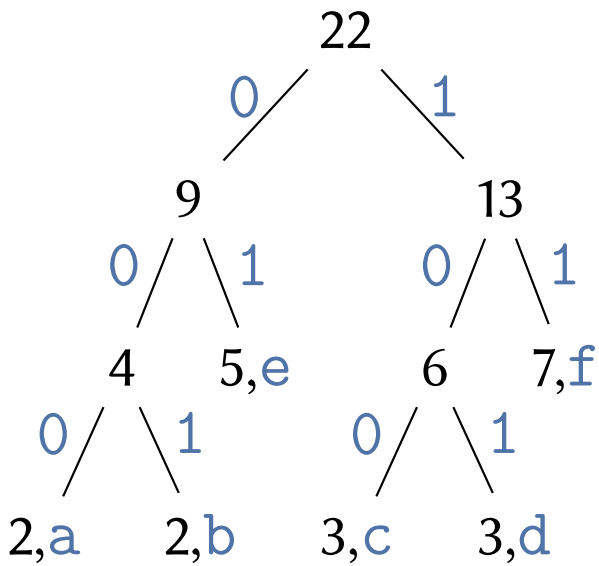
- alle Symbole an den
- Kanten auf diesem Weg

$h(\mathtt{d}) = \mathtt{101}$

Algorithmus für Huffman-Codes

Eingabe: $w =$ `afebfecaffdeddccefbeff`

Baum am Ende:



Codierung:

x	a	b	c	d	e	f
h(x)	000	001	100	101	01	11

Eigenschaften von Huffman-Codes

leicht zu dekodieren

- kein Codewort Anfangsstück eines anderen: *präfixfrei*
- einfache Dekodierung

Huffman-Code nicht eindeutig

- im allgemeinen mehrere Möglichkeiten, welche zwei Knoten vereinigt werden
- im Baum links und rechts vertauschbar

aber alle sind „gleich gut“:

- Unter allen präfixfreien Codes führen Huffman-Codes zu kürzesten Codierungen *des Wortes, für das die Huffman-Codierung konstruiert wurde.*

Block-Codierungen

Verallgemeinerung des obigen Verfahrens:

- Betrachte nicht Häufigkeiten einzelner Symbole,
- sondern für Teilwörter einer festen Länge $b > 1$.
- einziger Unterschied: an den Blättern des Huffman-Baumes stehen Wörter der Länge b.

kann bei Kompression helfen

- Beispiel:
  ca|ca|ca|bb|ca|ab|bb|ca|bb|bb|ca|ca|ca|ab|ca|ca|bb

x	ab	bb	ca
h(x)	00	01	1

Das sollten Sie mitnehmen

Informatik braucht Mathematik

Algorithmen brauchen Abstraktion

Präzision wichtig

- Verständlichkeit
- Verarbeitbarkeit durch Rechner
- Beweise

Herzlichen Dank für Ihre Aufmerksamkeit