Vorkurs **Datenstrukturen**

Sommersemester 2011

Vorkurs **Datenstrukturen**

Sommersemester 2011

Herzlich willkommen!

Abstrakte Datentypen und Datenstrukturen

- Ein abstrakter Datentyp besteht aus einer Sammlung von Operationen auf einer Menge von Objekten.
- Eine Datenstruktur implementiert einen abstrakten Datentyps.

Abstrakte Datentypen und Datenstrukturen

- Ein abstrakter Datentyp besteht aus einer Sammlung von Operationen auf einer Menge von Objekten.
- Eine Datenstruktur implementiert einen abstrakten Datentyps.

- Warum abstrakte Datentypen?
  - In Anwendungen treten dieselben Operationen für unterschiedlichste Datenmengen auf.
  - Gib eine Implementierung an!

Abstrakte Datentypen und Datenstrukturen

- Ein abstrakter Datentyp besteht aus einer Sammlung von Operationen auf einer Menge von Objekten.
- Eine Datenstruktur implementiert einen abstrakten Datentyps.

- Warum abstrakte Datentypen?
  - In Anwendungen treten dieselben Operationen für unterschiedlichste Datenmengen auf.
  - Gib eine Implementierung an!
- Zum Beispiel, füge Schlüssel ein und entferne Schlüssel geordnet nach ihrem Alter:

Abstrakte Datentypen und Datenstrukturen

- Ein abstrakter Datentyp besteht aus einer Sammlung von Operationen auf einer Menge von Objekten.
- Eine Datenstruktur implementiert einen abstrakten Datentyps.

- Warum abstrakte Datentypen?
  - In Anwendungen treten dieselben Operationen für unterschiedlichste Datenmengen auf.
  - Gib eine Implementierung an!
- Zum Beispiel, füge Schlüssel ein und entferne Schlüssel geordnet nach ihrem Alter:
  - Wenn der jüngste Schlüssel zu entfernen ist:
    Wähle die Datenstruktur

Abstrakte Datentypen und Datenstrukturen

- Ein abstrakter Datentyp besteht aus einer Sammlung von Operationen auf einer Menge von Objekten.
- Eine Datenstruktur implementiert einen abstrakten Datentyps.

- Warum abstrakte Datentypen?
  - In Anwendungen treten dieselben Operationen für unterschiedlichste Datenmengen auf.
  - Gib eine Implementierung an!
- Zum Beispiel, füge Schlüssel ein und entferne Schlüssel geordnet nach ihrem Alter:
  - Wenn der jüngste Schlüssel zu entfernen ist:
    Wähle die Datenstruktur „Stack“.

Abstrakte Datentypen und Datenstrukturen

- Ein abstrakter Datentyp besteht aus einer Sammlung von Operationen auf einer Menge von Objekten.
- Eine Datenstruktur implementiert einen abstrakten Datentyps.

- Warum abstrakte Datentypen?
  - In Anwendungen treten dieselben Operationen für unterschiedlichste Datenmengen auf.
  - Gib eine Implementierung an!
- Zum Beispiel, füge Schlüssel ein und entferne Schlüssel geordnet nach ihrem Alter:
  - Wenn der jüngste Schlüssel zu entfernen ist:
    Wähle die Datenstruktur „Stack".
  - Wenn der älteste Schlüssel zu entfernen ist:
    Wähle die Datenstruktur

Abstrakte Datentypen und Datenstrukturen

- Ein abstrakter Datentyp besteht aus einer Sammlung von Operationen auf einer Menge von Objekten.
- Eine Datenstruktur implementiert einen abstrakten Datentyps.

- Warum abstrakte Datentypen?
  - In Anwendungen treten dieselben Operationen für unterschiedlichste Datenmengen auf.
  - Gib eine Implementierung an!
- Zum Beispiel, füge Schlüssel ein und entferne Schlüssel geordnet nach ihrem Alter:
  - Wenn der jüngste Schlüssel zu entfernen ist:
Wähle die Datenstruktur „Stack“.
  - Wenn der älteste Schlüssel zu entfernen ist:
Wähle die Datenstruktur „Schlange“.

Abstrakte Datentypen und Datenstrukturen

- Ein abstrakter Datentyp besteht aus einer Sammlung von Operationen auf einer Menge von Objekten.
- Eine Datenstruktur implementiert einen abstrakten Datentyps.

- Warum abstrakte Datentypen?
  - In Anwendungen treten dieselben Operationen für unterschiedlichste Datenmengen auf.
  - Gib eine Implementierung an!
- Zum Beispiel, füge Schlüssel ein und entferne Schlüssel geordnet nach ihrem Alter:
  - Wenn der jüngste Schlüssel zu entfernen ist: Wähle die Datenstruktur „Stack“.
  - Wenn der älteste Schlüssel zu entfernen ist: Wähle die Datenstruktur „Schlange“.
  - Wenn Schlüssel mit zugeordneter Prioritäten eingefügt werden und der Schlüssel mit jeweils höchster Priorität zu entfernen ist: Benutze die Datenstruktur

Abstrakte Datentypen und Datenstrukturen

- Ein abstrakter Datentyp besteht aus einer Sammlung von Operationen auf einer Menge von Objekten.
- Eine Datenstruktur implementiert einen abstrakten Datentyps.

- Warum abstrakte Datentypen?
  - In Anwendungen treten dieselben Operationen für unterschiedlichste Datenmengen auf.
  - Gib eine Implementierung an!
- Zum Beispiel, füge Schlüssel ein und entferne Schlüssel geordnet nach ihrem Alter:
  - Wenn der jüngste Schlüssel zu entfernen ist:
    Wähle die Datenstruktur „Stack“.
  - Wenn der älteste Schlüssel zu entfernen ist:
    Wähle die Datenstruktur „Schlange“.
  - Wenn Schlüssel mit zugeordneter Prioritäten eingefügt werden und der Schlüssel mit jeweils höchster Priorität zu entfernen ist:
    Benutze die Datenstruktur „Heap“.

Effizienz I

Das Ziel der Vorlesung „**Datenstrukturen**“:

> Entwerfe eine möglichst effiziente Datenstruktur für einen gegebenen abstrakten Datentyp.

Effizienz I

Das Ziel der Vorlesung „**Datenstrukturen**“:

> Entwerfe eine möglichst effiziente Datenstruktur für einen gegebenen abstrakten Datentyp.

- Wie skaliert die Laufzeit einer Implementierung mit wachsender Eingabelänge?
  - Das Verhalten für große Eingabelängen ist kritisch.

Effizienz I

Das Ziel der Vorlesung „**Datenstrukturen**“:

> Entwerfe eine möglichst effiziente Datenstruktur für einen gegebenen abstrakten Datentyp.

- Wie skaliert die Laufzeit einer Implementierung mit wachsender Eingabelänge?
  - Das Verhalten für große Eingabelängen ist kritisch.
  - Für **Eingabelänge n** sei $T(n)$ die maximale Laufzeit für eine Eingabe der Länge n.

Effizienz I

Das Ziel der Vorlesung „**Datenstrukturen**“:

> Entwerfe eine möglichst effiziente Datenstruktur für einen gegebenen abstrakten Datentyp.

- Wie skaliert die Laufzeit einer Implementierung mit wachsender Eingabelänge?
  - Das Verhalten für große Eingabelängen ist kritisch.
  - Für **Eingabelänge n** sei $T(n)$ die maximale Laufzeit für eine Eingabe der Länge n.
- Führe eine asymptotische Analyse von Laufzeit und Speicherplatzverbrauch durch.

Effizienz I

Das Ziel der Vorlesung „**Datenstrukturen**“:

Entwerfe eine möglichst effiziente Datenstruktur für einen gegebenen abstrakten Datentyp.

- Wie skaliert die Laufzeit einer Implementierung mit wachsender Eingabelänge?
  - Das Verhalten für große Eingabelängen ist kritisch.
  - Für **Eingabelänge n** sei $T(n)$ die maximale Laufzeit für eine Eingabe der Länge n.
- Führe eine asymptotische Analyse von Laufzeit und Speicherplatzverbrauch durch.
  - Die Laufzeit ist linear, falls

Effizienz I

Das Ziel der Vorlesung „**Datenstrukturen**“:

Entwerfe eine möglichst effiziente Datenstruktur für einen gegebenen abstrakten Datentyp.

- Wie skaliert die Laufzeit einer Implementierung mit wachsender Eingabelänge?
  - Das Verhalten für große Eingabelängen ist kritisch.
  - Für **Eingabelänge n** sei $T(n)$ die maximale Laufzeit für eine Eingabe der Länge n.
- Führe eine asymptotische Analyse von Laufzeit und Speicherplatzverbrauch durch.
  - Die Laufzeit ist linear, falls $0 < \lim_{n\to\infty} \frac{T(n)}{n} < \infty$
  - und quadratisch, falls

Effizienz I

Das Ziel der Vorlesung „**Datenstrukturen**“:

Entwerfe eine möglichst effiziente Datenstruktur für einen gegebenen abstrakten Datentyp.

- Wie skaliert die Laufzeit einer Implementierung mit wachsender Eingabelänge?
  - Das Verhalten für große Eingabelängen ist kritisch.
  - Für **Eingabelänge n** sei $T(n)$ die maximale Laufzeit für eine Eingabe der Länge n.
- Führe eine asymptotische Analyse von Laufzeit und Speicherplatzverbrauch durch.
  - Die Laufzeit ist linear, falls $0 < \lim_{n\to\infty} \frac{T(n)}{n} < \infty$
  - und quadratisch, falls $0 < \lim_{n\to\infty} \frac{T(n)}{n^2} < \infty$.

Wir müssen uns mit Grenzwerten beschäftigen.

Effizienz II

Angenommen, wir möchten n Zahlen sortieren.

Effizienz II

Angenommen, wir möchten n Zahlen sortieren.

- Wir definieren die Eingabelänge als die Anzahl n der Zahlen.
- Ein Sortierverfahren wie Bubble Sort benötigt (ungefähr) quadratisch viele Vergleiche im schlimmsten Fall.

Effizienz II

Angenommen, wir möchten n Zahlen sortieren.

- Wir definieren die Eingabelänge als die Anzahl n der Zahlen.
- Ein Sortierverfahren wie Bubble Sort benötigt (ungefähr) quadratisch viele Vergleiche im schlimmsten Fall.
- Merge Sort hingegen kommt für jede Eingabe mit (ungefähr) $n \cdot \log_2 n$ Vergleichen aus.

Effizienz II

Angenommen, wir möchten n Zahlen sortieren.

- Wir definieren die Eingabelänge als die Anzahl n der Zahlen.
- Ein Sortierverfahren wie Bubble Sort benötigt (ungefähr) quadratisch viele Vergleiche im schlimmsten Fall.
- Merge Sort hingegen kommt für jede Eingabe mit (ungefähr) $n \cdot \log_2 n$ Vergleichen aus.

Die asymptotische Analyse zeigt:

$$\lim_{n \to \infty} \frac{n \cdot \log_2 n}{n^2} = 0$$

Merge Sort ist wesentlich schneller als Bubble Sort.

Wie schnell dominiert die Asymptotik?

Annahme: Ein einfacher Befehl benötigt 10^{-9} Sekunden.

Wie schnell dominiert die Asymptotik?

Annahme: Ein einfacher Befehl benötigt 10^{-9} Sekunden.

n	n^2	n^3	n^{10}	2^n	$n!$
16	256	4.096	$\geq 10^{12}$	65536	$\geq 10^{13}$
32	1.024	32.768	$\geq 10^{15}$	$\geq 4 \cdot 10^{9}$	$\geq 10^{31}$
64	4.096	262.144	$\geq 10^{18}$	$\geq 6 \cdot 10^{19}$	mehr als 10^{14} Jahre
128	16.384	2.097.152	mehr als 10 Jahre	mehr als 600 Jahre	
256	65.536	16.777.216			
512	262.144	134.217.728			
1024	1.048.576	$\geq 10^{9}$			
Million	$\geq 10^{12}$	$\geq 10^{18}$			
	mehr als 15 Minuten	mehr als 10 Jahre			

Effizienz III

Angenommen, wir möchten feststellen, ob eine Zahl x in einem sortierten Array von n Zahlen vorkommt.

Effizienz III

Angenommen, wir möchten feststellen, ob eine Zahl x in einem sortierten Array von n Zahlen vorkommt.

- Wir definieren die Eingabelänge wieder als die Anzahl n der Zahlen.
- Lineare Suche: Wir durchlaufen das Array von „links nach rechts“ und überprüfen jeweils ob wir x gefunden haben.
  - Mindestens n Vergleiche sind notwendig, wenn x nicht vorkommt.

Effizienz III

Angenommen, wir möchten feststellen, ob eine Zahl x in einem sortierten Array von n Zahlen vorkommt.

- Wir definieren die Eingabelänge wieder als die Anzahl n der Zahlen.
- Lineare Suche: Wir durchlaufen das Array von „links nach rechts“ und überprüfen jeweils ob wir x gefunden haben.

  Mindestens n Vergleiche sind notwendig, wenn x nicht vorkommt.
- Binärsuche vergleicht x mit $A[n/2]$.
  - Wenn $x = A[n/2]$, dann haben wir x gefunden!
  - Wenn $x < A[n/2]$, dann suche in der linken Hälfte $(A[1], \ldots, A[n/2] - 1)$,
  - und sonst in rechten Hälfte $(A[n/2+1], \ldots, A[n])$.

Effizienz III

Angenommen, wir möchten feststellen, ob eine Zahl x in einem sortierten Array von n Zahlen vorkommt.

- Wir definieren die Eingabelänge wieder als die Anzahl n der Zahlen.
- Lineare Suche: Wir durchlaufen das Array von „links nach rechts“ und überprüfen jeweils ob wir x gefunden haben.

  Mindestens n Vergleiche sind notwendig, wenn x nicht vorkommt.
- Binärsuche vergleicht x mit $A[n/2]$.
  - Wenn $x = A[n/2]$, dann haben wir x gefunden!
  - Wenn $x < A[n/2]$, dann suche in der linken Hälfte $(A[1], \ldots, A[n/2] - 1)$,
  - und sonst in rechten Hälfte $(A[n/2+1], \ldots, A[n])$.

Wer ist schneller: Lineare Suche oder Binärsuche?

Binärsuche

```
void Suche( int unten, int oben){
  if (oben < unten)
    std::cout « x « " wurde nicht gefunden."
     « std::endl;
```

Binärsuche

```
void Suche( int unten, int oben){
  if (oben < unten)
    std::cout « x « " wurde nicht gefunden."
     « std::endl;
  int mitte = (unten+oben)/2;
```

Binärsuche

```
void Suche( int unten, int oben){
  if (oben < unten)
    std::cout « x « " wurde nicht gefunden."
     « std::endl;
  int mitte = (unten+oben)/2;
  if (A[mitte] == x)
```

Binärsuche

```
void Suche( int unten, int oben){
  if (oben < unten)
    std::cout « x « " wurde nicht gefunden."
     « std::endl;
  int mitte = (unten+oben)/2;
  if (A[mitte] == x)
    std::cout « x « " wurde in Position "
     « mitte « " gefunden." « std::endl;
```

Binärsuche

```
void Suche( int unten, int oben){
  if (oben < unten)
    std::cout « x « " wurde nicht gefunden."
     « std::endl;
  int mitte = (unten+oben)/2;
  if (A[mitte] == x)
    std::cout « x « " wurde in Position "
     « mitte « " gefunden." « std::endl;
    else {
```

Binärsuche

```
void Suche( int unten, int oben){
   if (oben < unten)
     std::cout « x « " wurde nicht gefunden."
      « std::endl;
   int mitte = (unten+oben)/2;
   if (A[mitte] == x)
     std::cout « x « " wurde in Position "
      « mitte « " gefunden." « std::endl;
     else {
       if (x < A[mitte])
         Suche(unten, mitte-1);
       else
         Suche(mitte+1,oben);
     }
}
```

Rekursion: Eine fundamentale Programmiertechnik

- Wie kann man ein rekursives Programm verifizieren?
  Wie lassen sich Aussagen über die Laufzeit machen?
- Mit Hilfe der vollständigen Induktion.

Warum ist `Suche(a,b)` korrekt?

Wir beweisen Korrektheit durch Induktion nach der Anzahl $n = oben - unten + 1$ der Zahlen.

Warum ist `Suche(a,b)` korrekt?

Wir beweisen Korrektheit durch Induktion nach der Anzahl $n = oben - unten + 1$ der Zahlen.

- Der **Basisschritt** für $n \leq 0$:
  - Wenn $n \leq 0$, dann ist das Array leer und x kann nicht vorkommen.
  - `Suche(unten,oben)` ist für $n \leq 0$ korrekt.

Warum ist `Suche(a,b)` korrekt?

Wir beweisen Korrektheit durch Induktion nach der Anzahl $n = oben - unten + 1$ der Zahlen.

- Der **Basisschritt** für $n \leq 0$:
  - Wenn $n \leq 0$, dann ist das Array leer und x kann nicht vorkommen.
  - `Suche(unten,oben)` ist für $n \leq 0$ korrekt.
- Der **Induktionsschritt**:
  - Wir können die **Induktionsvoraussetzung** annehmen: `Suche(unten',oben')` ist korrekt, wenn $oben' - unten' \leq n$.
  - Wir müssen zeigen, dass `Suche(unten,oben)` für $oben - unten = n + 1$ korrekt ist.

Der Induktionsschritt

- Wenn an der Position `mitte` die Zahl x steht, wird sie ordnungsgemäß gefunden.
- Andernfalls ruft sich der Algorithmus rekursiv auf einem Teilproblem auf.

Der Induktionsschritt

- Wenn an der Position `mitte` die Zahl x steht, wird sie ordnungsgemäß gefunden.
- Andernfalls ruft sich der Algorithmus rekursiv auf einem Teilproblem auf.
  - Das Teilproblem hat weniger als $\mathtt{oben} - \mathtt{unten} - 1 = n$ Zahlen.

Der Induktionsschritt

- Wenn an der Position `mitte` die Zahl x steht, wird sie ordnungsgemäß gefunden.
- Andernfalls ruft sich der Algorithmus rekursiv auf einem Teilproblem auf.
  - Das Teilproblem hat weniger als $\texttt{oben} - \texttt{unten} - 1 = n$ Zahlen.
  - Auf einem Problem dieser Größe arbeitet der Algorithmus jedoch nach Induktionsannahme korrekt!

Wieviele Vergleiche maximal führt `suche(unten,oben)` aus?

Die Anzahl der Vergleiche hängt nur von der Zahl $n = oben - unten + 1$ der Elemente des Arrays ab.

Wieviele Vergleiche maximal führt `suche(unten,oben)` aus?

Die Anzahl der Vergleiche hängt nur von der Zahl $n = oben - unten + 1$ der Elemente des Arrays ab.

- Es gelte $n = 2^k - 1$.

Wieviele Vergleiche maximal führt `suche(unten,oben)` aus?

Die Anzahl der Vergleiche hängt nur von der Zahl $n = oben - unten + 1$ der Elemente des Arrays ab.

- Es gelte $n = 2^k - 1$.
  - Sei $T(n)$ die maximale Vergleichszahl für Arrays der Länge n.
  - Dann ist $T(1) =$

Wieviele Vergleiche maximal führt `suche(unten,oben)` aus?

Die Anzahl der Vergleiche hängt nur von der Zahl $n = oben - unten + 1$ der Elemente des Arrays ab.

- Es gelte $n = 2^k - 1$.
  - Sei $T(n)$ die maximale Vergleichszahl für Arrays der Länge n.
  - Dann ist $T(1) = 1$.

Wieviele Vergleiche maximal führt `suche(unten,oben)` aus?

Die Anzahl der Vergleiche hängt nur von der Zahl $n = oben - unten + 1$ der Elemente des Arrays ab.

- Es gelte $n = 2^k - 1$.
  - Sei $T(n)$ die maximale Vergleichszahl für Arrays der Länge n.
  - Dann ist $T(1) = 1$.
  - `Suche(unten,b oben)` arbeitet entweder in der linken Hälfte oder in der rechten Hälfte rekursiv weiter:
    - ★ Beide Hälften bestehen aus genau $2^{k-1} - 1$ Zahlen.

Wieviele Vergleiche maximal führt `suche(unten,oben)` aus?

Die Anzahl der Vergleiche hängt nur von der Zahl $n = oben - unten + 1$ der Elemente des Arrays ab.

- Es gelte $n = 2^k - 1$.
  - Sei $T(n)$ die maximale Vergleichszahl für Arrays der Länge n.
  - Dann ist $T(1) = 1$.
  - `Suche(unten,b oben)` arbeitet entweder in der linken Hälfte oder in der rechten Hälfte rekursiv weiter:
    - ★ Beide Hälften bestehen aus genau $2^{k-1} - 1$ Zahlen.
  - Es ist $T(2^k - 1) = T(2^{k-1} - 1) + 1$, falls $k \geq 1$.

Wieviele Vergleiche maximal führt `suche(unten,oben)` aus?

Die Anzahl der Vergleiche hängt nur von der Zahl $n = oben - unten + 1$ der Elemente des Arrays ab.

- Es gelte $n = 2^k - 1$.
  - Sei $T(n)$ die maximale Vergleichszahl für Arrays der Länge n.
  - Dann ist $T(1) = 1$.
  - `Suche(unten,b oben)` arbeitet entweder in der linken Hälfte oder in der rechten Hälfte rekursiv weiter:
    - Beide Hälften bestehen aus genau $2^{k-1} - 1$ Zahlen.
  - Es ist $T(2^k - 1) = T(2^{k-1} - 1) + 1$, falls $k \geq 1$.
  - Es ist $T(2^k - 1) =$

Wieviele Vergleiche maximal führt `suche(unten,oben)` aus?

Die Anzahl der Vergleiche hängt nur von der Zahl $n = oben - unten + 1$ der Elemente des Arrays ab.

- Es gelte $n = 2^k - 1$.
  - Sei $T(n)$ die maximale Vergleichszahl für Arrays der Länge n.
  - Dann ist $T(1) = 1$.
  - `Suche(unten,b oben)` arbeitet entweder in der linken Hälfte oder in der rechten Hälfte rekursiv weiter:
    - ★ Beide Hälften bestehen aus genau $2^{k-1} - 1$ Zahlen.
  - Es ist $T(2^k - 1) = T(2^{k-1} - 1) + 1$, falls $k \geq 1$.
  - Es ist $T(2^k - 1) = k$. Warum?

Wieviele Vergleiche maximal führt `suche(unten,oben)` aus?

Die Anzahl der Vergleiche hängt nur von der Zahl $n = oben - unten + 1$ der Elemente des Arrays ab.

- Es gelte $n = 2^k - 1$.
  - Sei $T(n)$ die maximale Vergleichszahl für Arrays der Länge n.
  - Dann ist $T(1) = 1$.
  - `Suche(unten,b oben)` arbeitet entweder in der linken Hälfte oder in der rechten Hälfte rekursiv weiter:
    - Beide Hälften bestehen aus genau $2^{k-1} - 1$ Zahlen.
  - Es ist $T(2^k - 1) = T(2^{k-1} - 1) + 1,$ falls $k \geq 1$.
  - Es ist $T(2^k - 1) = k$. Warum?
- Und wenn n eine beliebige natürliche Zahl ist? Dann ist

$$T(n) =$$

Wieviele Vergleiche maximal führt `suche(unten,oben)` aus?

Die Anzahl der Vergleiche hängt nur von der Zahl $n = oben - unten + 1$ der Elemente des Arrays ab.

- Es gelte $n = 2^k - 1$.
  - Sei $T(n)$ die maximale Vergleichszahl für Arrays der Länge n.
  - Dann ist $T(1) = 1$.
  - `Suche(unten,b oben)` arbeitet entweder in der linken Hälfte oder in der rechten Hälfte rekursiv weiter:
    - ★ Beide Hälften bestehen aus genau $2^{k-1} - 1$ Zahlen.
  - Es ist $T(2^k - 1) = T(2^{k-1} - 1) + 1$, falls $k \geq 1$.
  - Es ist $T(2^k - 1) = k$. Warum?
- Und wenn n eine beliebige natürliche Zahl ist? Dann ist

$$T(n) = \lceil \log_2(n+1) \rceil.$$

Binärsuche ist viel schneller als lineare Suche, denn $\lim_{n\to\infty} \frac{\log_2 n}{n} = 0$.

Der Fahrplan

Wir machen mit

rekursiver Programmierung / vollständiger Induktion

weiter und kehren dann zur

asymptotischen Analyse

zurück.

Die Türme von Hanoi

- Die **Anfangskonfiguration**:
  - Wir haben drei Stäbe 1, 2 und 3.
  - Ursprünglich besitzt Stab 1 N Ringe, wobei die Ringe in absteigender Größe auf dem Stab aufgereiht sind:
    - Der größte Ring von Stab 1 ist also der unterste Ring.
  - Die Stäbe 2 und 3 sind zu Anfang leer.

Die Türme von Hanoi

- Die **Anfangskonfiguration**:
  - Wir haben drei Stäbe 1, 2 und 3.
  - Ursprünglich besitzt Stab 1 N Ringe, wobei die Ringe in absteigender Größe auf dem Stab aufgereiht sind:
    - Der größte Ring von Stab 1 ist also der unterste Ring.
  - Die Stäbe 2 und 3 sind zu Anfang leer.
- Die **Züge**
  - Bewege einen zuoberst liegenden Ring von einem Stab zu einem anderen.
  - Der Zug ist nur dann erlaubt, wenn der Ring auf einen größeren Ring gelegt wird oder wenn der Stab leer ist.

Die Türme von Hanoi

- Die **Anfangskonfiguration**:
  - Wir haben drei Stäbe 1, 2 und 3.
  - Ursprünglich besitzt Stab 1 N Ringe, wobei die Ringe in absteigender Größe auf dem Stab aufgereiht sind:
    - Der größte Ring von Stab 1 ist also der unterste Ring.
  - Die Stäbe 2 und 3 sind zu Anfang leer.
- Die **Züge**
  - Bewege einen zuoberst liegenden Ring von einem Stab zu einem anderen.
  - Der Zug ist nur dann erlaubt, wenn der Ring auf einen größeren Ring gelegt wird oder wenn der Stab leer ist.
- Die **Zielkonfiguration**:
  - Alle Ringe müssen nach Stab 2 bewegt werden.

Ein rekursives Programm

```
void Hanoi( int N, int stab1, int stab2, int stab3)
```

Ein rekursives Programm

```
void Hanoi( int N, int stab1, int stab2, int stab3)
{if (N==1)
```

Ein rekursives Programm

```
void Hanoi( int N, int stab1, int stab2, int stab3)
{if (N==1)
    bewege einen Ring von Stab stab1 nach Stab stab2;
else {
```

Ein rekursives Programm

```
void Hanoi( int N, int stab1, int stab2, int stab3)
{if (N==1)
    bewege einen Ring von Stab stab1 nach Stab stab2;
else {
    Hanoi(N-1,stab1,stab3,stab2);
```

Ein rekursives Programm

```
void Hanoi( int N, int stab1, int stab2, int stab3)
{if (N==1)
    bewege einen Ring von Stab stab1 nach Stab stab2;
else {
    Hanoi(N-1,stab1,stab3,stab2);
    bewege einen Ring von Stab stab1 nach Stab stab2;
```

Ein rekursives Programm

```
void Hanoi( int N, int stab1, int stab2, int stab3)
{if (N==1)
    bewege einen Ring von Stab stab1 nach Stab stab2;
else {
    Hanoi(N-1,stab1,stab3,stab2);
    bewege einen Ring von Stab stab1 nach Stab stab2;
    Hanoi(N-1,stab3,stab2,stab1); }}
```

Ein rekursives Programm

```
void Hanoi( int N, int stab1, int stab2, int stab3)
{if (N==1)
    bewege einen Ring von Stab stab1 nach Stab stab2;
else {
    Hanoi(N-1,stab1,stab3,stab2);
    bewege einen Ring von Stab stab1 nach Stab stab2;
    Hanoi(N-1,stab3,stab2,stab1); }}
```

- Wie zeigt man, dass `Hanoi` korrekt ist?

Ein rekursives Programm

```
void Hanoi( int N, int stab1, int stab2, int stab3)
{if (N==1)
    bewege einen Ring von Stab stab1 nach Stab stab2;
else {
    Hanoi(N-1,stab1,stab3,stab2);
    bewege einen Ring von Stab stab1 nach Stab stab2;
    Hanoi(N-1,stab3,stab2,stab1); }}
```

- Wie zeigt man, dass `Hanoi` korrekt ist?
- Sei $T(N)$ die Anzahl der Ringbewegungen nach Aufruf des Programms Hanoi(N,*,*,*): Wie bestimmt man $T(N)$?

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n =$$

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$:

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$.

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n+1$:

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i =$

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n + 1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) =$

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n + 1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n \cdot (n+1)}{2} + (n+1) =$

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n \cdot (n+1)}{2} + (n+1) = \frac{(n+1) \cdot (n+2)}{2}$.

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0\cdot(0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n\cdot(n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n\cdot(n+1)}{2} + (n+1) = \frac{(n+1)\cdot(n+2)}{2}$. Stimmt!

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n \cdot (n+1)}{2} + (n+1) = \frac{(n+1) \cdot (n+2)}{2}$. Stimmt!
- Ein direktes Argument:

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0\cdot(0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n\cdot(n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n\cdot(n+1)}{2} + (n+1) = \frac{(n+1)\cdot(n+2)}{2}$. Stimmt!
- Ein direktes Argument:
  - Betrachte ein Gitter mit n Zeilen und n Spalten:

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n + 1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n \cdot (n+1)}{2} + (n+1) = \frac{(n+1) \cdot (n+2)}{2}$. Stimmt!
- Ein direktes Argument:
  - Betrachte ein Gitter mit n Zeilen und n Spalten: n^2 Gitterpunkte.

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0\cdot(0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n\cdot(n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n\cdot(n+1)}{2} + (n+1) = \frac{(n+1)\cdot(n+2)}{2}$. Stimmt!
- Ein direktes Argument:
  - Betrachte ein Gitter mit n Zeilen und n Spalten: n^2 Gitterpunkte.
  - Wir müssen die Gitterpunkte unterhalb der Hauptdiagonale und auf der Hauptdiagonale zählen.

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0\cdot(0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n\cdot(n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n\cdot(n+1)}{2} + (n+1) = \frac{(n+1)\cdot(n+2)}{2}$. Stimmt!
- Ein direktes Argument:
  - Betrachte ein Gitter mit n Zeilen und n Spalten: n^2 Gitterpunkte.
  - Wir müssen die Gitterpunkte unterhalb der Hauptdiagonale und auf der Hauptdiagonale zählen.
    - Die Hauptdiagonale besitzt n Gitterpunkte.

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n \cdot (n+1)}{2} + (n+1) = \frac{(n+1) \cdot (n+2)}{2}$. Stimmt!
- Ein direktes Argument:
  - Betrachte ein Gitter mit n Zeilen und n Spalten: n^2 Gitterpunkte.
  - Wir müssen die Gitterpunkte unterhalb der Hauptdiagonale und auf der Hauptdiagonale zählen.
    - Die Hauptdiagonale besitzt n Gitterpunkte.
    - Unterhalb der Hauptdiagonale befindet sich die Hälfte der verbleibenden $n^2 - n$ Gitterpunkte.

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n \cdot (n+1)}{2} + (n+1) = \frac{(n+1) \cdot (n+2)}{2}$. Stimmt!
- Ein direktes Argument:
  - Betrachte ein Gitter mit n Zeilen und n Spalten: n^2 Gitterpunkte.
  - Wir müssen die Gitterpunkte unterhalb der Hauptdiagonale und auf der Hauptdiagonale zählen.
    - Die Hauptdiagonale besitzt n Gitterpunkte.
    - Unterhalb der Hauptdiagonale befindet sich die Hälfte der verbleibenden $n^2 - n$ Gitterpunkte.
  - Also folgt $\sum_{i=1}^{n} i =$

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n + 1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n \cdot (n+1)}{2} + (n+1) = \frac{(n+1) \cdot (n+2)}{2}$. Stimmt!
- Ein direktes Argument:
  - Betrachte ein Gitter mit n Zeilen und n Spalten: n^2 Gitterpunkte.
  - Wir müssen die Gitterpunkte unterhalb der Hauptdiagonale und auf der Hauptdiagonale zählen.
    - Die Hauptdiagonale besitzt n Gitterpunkte.
    - Unterhalb der Hauptdiagonale befindet sich die Hälfte der verbleibenden $n^2 - n$ Gitterpunkte.
  - Also folgt $\sum_{i=1}^{n} i = n +$

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n + 1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n \cdot (n+1)}{2} + (n+1) = \frac{(n+1) \cdot (n+2)}{2}$. Stimmt!
- Ein direktes Argument:
  - Betrachte ein Gitter mit n Zeilen und n Spalten: n^2 Gitterpunkte.
  - Wir müssen die Gitterpunkte unterhalb der Hauptdiagonale und auf der Hauptdiagonale zählen.
    - Die Hauptdiagonale besitzt n Gitterpunkte.
    - Unterhalb der Hauptdiagonale befindet sich die Hälfte der verbleibenden $n^2 - n$ Gitterpunkte.
  - Also folgt $\sum_{i=1}^{n} i = n + \frac{n^2 - n}{2}$

Die Summe der ersten n Zahlen

$$\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n \cdot (n+1)}{2}.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} i = 0$ und $\frac{0 \cdot (0+1)}{2} = 0$. Stimmt!
  - Induktionsschritt von n auf $n + 1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.
    - $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n \cdot (n+1)}{2} + (n+1) = \frac{(n+1) \cdot (n+2)}{2}$. Stimmt!
- Ein direktes Argument:
  - Betrachte ein Gitter mit n Zeilen und n Spalten: n^2 Gitterpunkte.
  - Wir müssen die Gitterpunkte unterhalb der Hauptdiagonale und auf der Hauptdiagonale zählen.
    - Die Hauptdiagonale besitzt n Gitterpunkte.
    - Unterhalb der Hauptdiagonale befindet sich die Hälfte der verbleibenden $n^2 - n$ Gitterpunkte.
  - Also folgt $\sum_{i=1}^{n} i = n + \frac{n^2 - n}{2} = \frac{n \cdot (n+1)}{2}$.

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1} \text{ falls } a \neq 1.$$

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$:

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} a^i = 1$

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$.

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$. Stimmt!

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$. Stimmt!
  - Induktionsschritt von n auf $n + 1$:

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ gilt.

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ gilt. Dann ist
    - $\sum_{i=1}^{n+1} a^i$

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$. Stimmt!
  - Induktionsschritt von n auf $n + 1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ gilt. Dann ist
    - $\sum_{i=1}^{n+1} a^i = \sum_{i=1}^{n} a^i + a^{n+1}$

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$. Stimmt!
  - Induktionsschritt von n auf $n + 1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ gilt. Dann ist
    - $\sum_{i=1}^{n+1} a^i = \sum_{i=1}^{n} a^i + a^{n+1} = \frac{a^{n+1}-1}{a-1} + a^{n+1}$

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$. Stimmt!
  - Induktionsschritt von n auf $n + 1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ gilt. Dann ist
    - $\sum_{i=1}^{n+1} a^i = \sum_{i=1}^{n} a^i + a^{n+1} = \frac{a^{n+1}-1}{a-1} + a^{n+1} = \frac{a^{n+2}-1}{a-1}$. Stimmt!

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ gilt. Dann ist
    - $\sum_{i=1}^{n+1} a^i = \sum_{i=1}^{n} a^i + a^{n+1} = \frac{a^{n+1}-1}{a-1} + a^{n+1} = \frac{a^{n+2}-1}{a-1}$. Stimmt!
- Ein direktes Argument:

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n = 0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ gilt. Dann ist
    - $\sum_{i=1}^{n+1} a^i = \sum_{i=1}^{n} a^i + a^{n+1} = \frac{a^{n+1}-1}{a-1} + a^{n+1} = \frac{a^{n+2}-1}{a-1}$. Stimmt!
- Ein direktes Argument:

$$(a-1) \cdot \sum_{i=0}^{n} a^i \quad = \quad a \cdot \sum_{i=0}^{n} a^i - \sum_{i=0}^{n} a^i$$

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n=0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ gilt. Dann ist
    - $\sum_{i=1}^{n+1} a^i = \sum_{i=1}^{n} a^i + a^{n+1} = \frac{a^{n+1}-1}{a-1} + a^{n+1} = \frac{a^{n+2}-1}{a-1}$. Stimmt!
- Ein direktes Argument:

$$\begin{aligned}(a-1)\cdot\sum_{i=0}^{n} a^i &= a\cdot\sum_{i=0}^{n} a^i - \sum_{i=0}^{n} a^i\\ &= \sum_{i=1}^{n+1} a^i - \sum_{i=0}^{n} a^i =\end{aligned}$$

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n=0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ gilt. Dann ist
    - $\sum_{i=1}^{n+1} a^i = \sum_{i=1}^{n} a^i + a^{n+1} = \frac{a^{n+1}-1}{a-1} + a^{n+1} = \frac{a^{n+2}-1}{a-1}$. Stimmt!
- Ein direktes Argument:

$$\begin{aligned}(a-1)\cdot\sum_{i=0}^{n} a^i &= a\cdot\sum_{i=0}^{n} a^i - \sum_{i=0}^{n} a^i \\ &= \sum_{i=1}^{n+1} a^i - \sum_{i=0}^{n} a^i = a^{n+1} - a^0\end{aligned}$$

Die geometrische Reihe

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1} \text{ falls } a \neq 1.$$

- Vollständige Induktion nach n:
  - Induktionsbasis für $n=0$: $\sum_{i=1}^{0} a^i = 1$ und $\frac{a^{0+1}-1}{a-1} = 1$. Stimmt!
  - Induktionsschritt von n auf $n+1$:
    - Wir können annnehmen, dass $\sum_{i=1}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ gilt. Dann ist
    - $\sum_{i=1}^{n+1} a^i = \sum_{i=1}^{n} a^i + a^{n+1} = \frac{a^{n+1}-1}{a-1} + a^{n+1} = \frac{a^{n+2}-1}{a-1}$. Stimmt!
- Ein direktes Argument:

$$\begin{aligned}(a-1)\cdot\sum_{i=0}^{n} a^i &= a\cdot\sum_{i=0}^{n} a^i - \sum_{i=0}^{n} a^i\\ &= \sum_{i=1}^{n+1} a^i - \sum_{i=0}^{n} a^i = a^{n+1} - a^0 = a^{n+1}-1\end{aligned}$$

Kombinatorische Fragestellungen

Die Menge U bestehe aus n Objekten.

Kombinatorische Fragestellungen

Die Menge U bestehe aus n Objekten.

(a) Zeige, dass U genau 2^n Teilmengen besitzt, wenn wir die leere Menge als eine Teilmenge von U zulassen.

Kombinatorische Fragestellungen

Die Menge U bestehe aus n Objekten.

(a) Zeige, dass U genau 2^n Teilmengen besitzt, wenn wir die leere Menge als eine Teilmenge von U zulassen.

(b) Zeige, dass U genau $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ Teilmengen mit k Elementen besitzt.

Kombinatorische Fragestellungen

Die Menge U bestehe aus n Objekten.

(a) Zeige, dass U genau 2^n Teilmengen besitzt, wenn wir die leere Menge als eine Teilmenge von U zulassen.

(b) Zeige, dass U genau $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ Teilmengen mit k Elementen besitzt.

(c) Zeige: Es gibt genau $n! = n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1$ verschiedene Permutationen von n Objekten.

Binomischer Lehrsatz

Zeige: $(a + b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$. (Zeige $\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}$ zuerst.)

Geometrische Fragestellungen

- Teile ein Rechteck durch Geraden in Teilflächen.
- Kann man die Teilflächen immer so mit den Farben Schwarz und Weiß färben, dass Teilflächen, die an einer Kante zusammenstoßen, verschiedene Farben besitzen?

Wo ist der Fehler?

Wir zeigen, dass je zwei natürliche Zahlen a und b gleich sind.

Wo ist der Fehler?

Wir zeigen, dass je zwei natürliche Zahlen a und b gleich sind.

- Wir setzen $k = \max\{a, b\}$ und führen eine Induktion nach k.

Wo ist der Fehler?

Wir zeigen, dass je zwei natürliche Zahlen a und b gleich sind.

- Wir setzen $k = \max\{a, b\}$ und führen eine Induktion nach k.
  - Im Basisschritt haben wir $k = 0$ und deshalb ist $a = 0 = b$ und das war zu zeigen.

Wo ist der Fehler?

Wir zeigen, dass je zwei natürliche Zahlen a und b gleich sind.

- Wir setzen $k = \max\{a, b\}$ und führen eine Induktion nach k.
  - Im Basisschritt haben wir $k = 0$ und deshalb ist $a = 0 = b$ und das war zu zeigen.
  - Im Induktionsschritt ist $\max\{a, b\} = k + 1$.
    - Wir können die Induktionsbehauptung auf $a - 1$ und $b - 1$ anwenden, denn $\max\{a - 1, b - 1\} = k$.

Wo ist der Fehler?

Wir zeigen, dass je zwei natürliche Zahlen a und b gleich sind.

- Wir setzen $k = \max\{a, b\}$ und führen eine Induktion nach k.
  - Im Basisschritt haben wir $k = 0$ und deshalb ist $a = 0 = b$ und das war zu zeigen.
  - Im Induktionsschritt ist $\max\{a, b\} = k + 1$.
    - Wir können die Induktionsbehauptung auf $a - 1$ und $b - 1$ anwenden, denn $\max\{a - 1, b - 1\} = k$.
    - Also ist $a - 1 = b - 1$ und die Behauptung $a = b$ folgt.

Wo ist der Fehler?

Wir zeigen, dass je zwei natürliche Zahlen a und b gleich sind.

- Wir setzen $k = \max\{a, b\}$ und führen eine Induktion nach k.
  - Im Basisschritt haben wir $k = 0$ und deshalb ist $a = 0 = b$ und das war zu zeigen.
  - Im Induktionsschritt ist $\max\{a, b\} = k + 1$.
    - Wir können die Induktionsbehauptung auf $a - 1$ und $b - 1$ anwenden, denn $\max\{a - 1, b - 1\} = k$.
    - Also ist $a - 1 = b - 1$ und die Behauptung $a = b$ folgt.
- Die Behauptung ist richtig für $k = 0$, aber falsch schon für $k = 1$.

Zurück zur Asymptotik

- Wir haben behauptet, dass Mergesort viel schneller als Bubblesort ist, weil

$$\lim_{n \to \infty} \frac{n \cdot \log_2 n}{n^2} = 0$$

  - Warum ist der Grenzwert 0?

Zurück zur Asymptotik

- Wir haben behauptet, dass Mergesort viel schneller als Bubblesort ist, weil

$$\lim_{n \to \infty} \frac{n \cdot \log_2 n}{n^2} = 0$$

  - Warum ist der Grenzwert 0?

- Wir haben behauptet, dass Binärsuche viel schneller als lineare Suche ist, weil

$$\lim_{n \to \infty} \frac{\log_2 n}{n} = 0$$

  - Warum ist auch dieser Grenzwert 0?

Zurück zur Asymptotik

- Wir haben behauptet, dass Mergesort viel schneller als Bubblesort ist, weil
$$\lim_{n\to\infty} \frac{n \cdot \log_2 n}{n^2} = 0$$
  - Warum ist der Grenzwert 0?
- Wir haben behauptet, dass Binärsuche viel schneller als lineare Suche ist, weil
$$\lim_{n\to\infty} \frac{\log_2 n}{n} = 0$$
  - Warum ist auch dieser Grenzwert 0?

- Wir untersuchen unendliche Folgen $(a_n) = a_1, a_2, \ldots, a_k \ldots$.
- Wann konvergiert (a_n) gegen einen Grenzwert a?

Man sagt, dass die Folge $(a_n) = a_0, a_1, \ldots$ gegen $a \in \mathbb{R}$ konvergiert, und nennt a dann den Grenzwert oder Limes der Folge (a_n), formal

$$\lim_{n \to \infty} a_n = a,$$

Grenzwerte I

Man sagt, dass die Folge $(a_n) = a_0, a_1, \ldots$ gegen $a \in \mathbb{R}$ konvergiert, und nennt a dann den Grenzwert oder Limes der Folge (a_n), formal

$$\lim_{n\to\infty} a_n = a,$$

wenn es für jedes $\varepsilon > 0$ ein $N \in \mathbb{N}$ gibt, so dass $|a_n - a| < \varepsilon$ für alle $n > N$ gilt.

Grenzwerte I

Man sagt, dass die Folge $(a_n) = a_0, a_1, \ldots$ gegen $a \in \mathbb{R}$ konvergiert, und nennt a dann den Grenzwert oder Limes der Folge (a_n), formal

$$\lim_{n \to \infty} a_n = a,$$

wenn es für jedes $\varepsilon > 0$ ein $N \in \mathbb{N}$ gibt, so dass $|a_n - a| < \varepsilon$ für alle $n > N$ gilt.

- **Zeige**: Jede Folge (a_n) besitzt höchstens einen Grenzwert.

Grenzwerte I

Man sagt, dass die Folge $(a_n) = a_0, a_1, \ldots$ gegen $a \in \mathbb{R}$ konvergiert, und nennt a dann den Grenzwert oder Limes der Folge (a_n), formal

$$\lim_{n\to\infty} a_n = a,$$

wenn es für jedes $\varepsilon > 0$ ein $N \in \mathbb{N}$ gibt, so dass $|a_n - a| < \varepsilon$ für alle $n > N$ gilt.

- **Zeige**: Jede Folge (a_n) besitzt höchstens einen Grenzwert.
- Warum gilt $\lim_{n\to\infty} \frac{1}{n} = 0$?
  - Für $\varepsilon > 0$ wähle $N =$

Grenzwerte I

Man sagt, dass die Folge $(a_n) = a_0, a_1, \ldots$ gegen $a \in \mathbb{R}$ konvergiert, und nennt a dann den Grenzwert oder Limes der Folge (a_n), formal

$$\lim_{n\to\infty} a_n = a,$$

wenn es für jedes $\varepsilon > 0$ ein $N \in \mathbb{N}$ gibt, so dass $|a_n - a| < \varepsilon$ für alle $n > N$ gilt.

- **Zeige**: Jede Folge (a_n) besitzt höchstens einen Grenzwert.
- Warum gilt $\lim_{n\to\infty} \frac{1}{n} = 0$?
  - Für $\varepsilon > 0$ wähle $N = \lceil \frac{1}{\varepsilon} \rceil$.
  - Dann gilt $|\frac{1}{n} - 0| = \frac{1}{n} < \varepsilon$ für $n > N$.

Grenzwerte I

Man sagt, dass die Folge $(a_n) = a_0, a_1, \ldots$ gegen $a \in \mathbb{R}$ konvergiert, und nennt a dann den Grenzwert oder Limes der Folge (a_n), formal

$$\lim_{n\to\infty} a_n = a,$$

wenn es für jedes $\varepsilon > 0$ ein $N \in \mathbb{N}$ gibt, so dass $|a_n - a| < \varepsilon$ für alle $n > N$ gilt.

- **Zeige**: Jede Folge (a_n) besitzt höchstens einen Grenzwert.
- Warum gilt $\lim_{n\to\infty} \frac{1}{n} = 0$?
  - Für $\varepsilon > 0$ wähle $N = \lceil \frac{1}{\varepsilon} \rceil$.
  - Dann gilt $|\frac{1}{n} - 0| = \frac{1}{n} < \varepsilon$ für $n > N$.
- Warum gilt $\lim_{n\to\infty} \frac{1}{n^{1/k}} = 0$? für jede natürliche Zahl k?

Grenzwerte II

- Wir sagen, dass eine Folge gegen „∞ konvergiert“, falls für jede Zahl $a \in \mathbb{R}$ fast alle Folgenglieder größer als a sind.

Grenzwerte II

- Wir sagen, dass eine Folge gegen „∞ konvergiert“, falls für jede Zahl $a \in \mathbb{R}$ fast alle Folgenglieder größer als a sind.
  - Konvergiert die Folge $a_n = \sum_{k=1}^{n} \frac{1}{k^2}$ gegen ∞?

- Wir sagen, dass eine Folge gegen „∞ konvergiert“, falls für jede Zahl $a \in \mathbb{R}$ fast alle Folgenglieder größer als a sind.
  - Konvergiert die Folge $a_n = \sum_{k=1}^{n} \frac{1}{k^2}$ gegen ∞?
  - Konvergiert die Folge $a_n = \sum_{k=1}^{n} \frac{1}{k}$ gegen ∞?

Grenzwerte II

- Wir sagen, dass eine Folge gegen „∞ konvergiert“, falls für jede Zahl $a \in \mathbb{R}$ fast alle Folgenglieder größer als a sind.
  - Konvergiert die Folge $a_n = \sum_{k=1}^{n} \frac{1}{k^2}$ gegen ∞?
  - Konvergiert die Folge $a_n = \sum_{k=1}^{n} \frac{1}{k}$ gegen ∞?

  Konvergenz gegen $-\infty$ wird analog definiert.

Grenzwerte II

- Wir sagen, dass eine Folge gegen „∞ konvergiert“, falls für jede Zahl $a \in \mathbb{R}$ fast alle Folgenglieder größer als a sind.
  - Konvergiert die Folge $a_n = \sum_{k=1}^{n} \frac{1}{k^2}$ gegen ∞?
  - Konvergiert die Folge $a_n = \sum_{k=1}^{n} \frac{1}{k}$ gegen ∞?

  Konvergenz gegen $-\infty$ wird analog definiert.
- Hat eine Folge keinen Grenzwert, so heißt sie divergent.
  - Für jede Zahl $a \in \mathbb{R}$ gibt es ein $\varepsilon > 0$, so dass unendlich viele Folgenglieder *außerhalb* der ε-Umgebung von a liegen.

Grenzwerte II

- Wir sagen, dass eine Folge gegen „∞ konvergiert", falls für jede Zahl $a \in \mathbb{R}$ fast alle Folgenglieder größer als a sind.
  - Konvergiert die Folge $a_n = \sum_{k=1}^{n} \frac{1}{k^2}$ gegen ∞?
  - Konvergiert die Folge $a_n = \sum_{k=1}^{n} \frac{1}{k}$ gegen ∞?

  Konvergenz gegen $-\infty$ wird analog definiert.
- Hat eine Folge keinen Grenzwert, so heißt sie divergent.
  - Für jede Zahl $a \in \mathbb{R}$ gibt es ein $\varepsilon > 0$, so dass unendlich viele Folgenglieder *außerhalb* der ε-Umgebung von a liegen.
  - Die alternierende Folge $a_n = (-1)^n$ divergiert, da sie unendlich oft zwischen 1 und -1 hin und her springt.

Das Einschließungs- und Majorantenkriterium

- Folgen (a_n) mit $\lim a_n = 0$ heißen **Nullfolgen.**

  Nullfolgen spielen eine Sonderrolle, denn $\lim a_n = a$ gilt genau dann, wenn $(a_n - a)$ eine Nullfolge ist.

Das Einschließungs- und Majorantenkriterium

- Folgen (a_n) mit $\lim a_n = 0$ heißen **Nullfolgen**.

  Nullfolgen spielen eine Sonderrolle, denn $\lim a_n = a$ gilt genau dann, wenn $(a_n - a)$ eine Nullfolge ist.
- Gilt $|a_n| \leq c \cdot |b_n|$ für eine Konstante c und fast alle n, dann ist die Folge (b_n) eine **Majorante** der Folge (a_n).

Das Einschließungs- und Majorantenkriterium

- Folgen (a_n) mit $\lim a_n = 0$ heißen **Nullfolgen**.

  Nullfolgen spielen eine Sonderrolle, denn $\lim a_n = a$ gilt genau dann, wenn $(a_n - a)$ eine Nullfolge ist.
- Gilt $|a_n| \leq c \cdot |b_n|$ für eine Konstante c und fast alle n, dann ist die Folge (b_n) eine **Majorante** der Folge (a_n).
- (a_n) heißt **beschränkt**, wenn die Folge $b_n = 1$ eine Majorante ist.

Das Einschließungs- und Majorantenkriterium

- Folgen (a_n) mit $\lim a_n = 0$ heißen **Nullfolgen**.

  Nullfolgen spielen eine Sonderrolle, denn $\lim a_n = a$ gilt genau dann, wenn $(a_n - a)$ eine Nullfolge ist.
- Gilt $|a_n| \leq c \cdot |b_n|$ für eine Konstante c und fast alle n, dann ist die Folge (b_n) eine **Majorante** der Folge (a_n).
- (a_n) heißt **beschränkt**, wenn die Folge $b_n = 1$ eine Majorante ist.

- Das Einschließungskriterium: Gilt $\lim a_n = \lim b_n = a$ und $a_n \leq x_n \leq b_n$ für fast alle n, so gilt auch $\lim x_n = a$.

Das Einschließungs- und Majorantenkriterium

- Folgen (a_n) mit $\lim a_n = 0$ heißen **Nullfolgen**.

  Nullfolgen spielen eine Sonderrolle, denn $\lim a_n = a$ gilt genau dann, wenn $(a_n - a)$ eine Nullfolge ist.
- Gilt $|a_n| \leq c \cdot |b_n|$ für eine Konstante c und fast alle n, dann ist die Folge (b_n) eine **Majorante** der Folge (a_n).
- (a_n) heißt **beschränkt**, wenn die Folge $b_n = 1$ eine Majorante ist.

- Das Einschließungskriterium: Gilt $\lim a_n = \lim b_n = a$ und $a_n \leq x_n \leq b_n$ für fast alle n, so gilt auch $\lim x_n = a$.

  Zeige, dass $\lim \frac{1}{n^k} = 0$ für jedes $k \geq 1$ gilt.

Das Einschließungs- und Majorantenkriterium

- Folgen (a_n) mit $\lim a_n = 0$ heißen **Nullfolgen**.

  Nullfolgen spielen eine Sonderrolle, denn $\lim\, a_n = a$ gilt genau dann, wenn $(a_n - a)$ eine Nullfolge ist.
- Gilt $|a_n| \leq c \cdot |b_n|$ für eine Konstante c und fast alle n, dann ist die Folge (b_n) eine **Majorante** der Folge (a_n).
- (a_n) heißt **beschränkt**, wenn die Folge $b_n = 1$ eine Majorante ist.

- Das Einschließungskriterium: Gilt $\lim\, a_n = \lim\, b_n = a$ und $a_n \leq x_n \leq b_n$ für fast alle n, so gilt auch $\lim\, x_n = a$.

  Zeige, dass $\lim\, \frac{1}{n^k} = 0$ für jedes $k \geq 1$ gilt.
- Das Majorantenkriterium für Nullfolgen: Hat die Folge (a_n) eine Nullfolge als ihre Majorante, so ist auch (a_n) eine Nullfolge.

Arithmetische Operationen auf Folgen

(a_n) und (b_n) seien konvergente Folgen mit den jeweiligen Grenzwerten a und b. Dann gilt:

1. $\lim (a_n \pm b_n) = a \pm b$;

Arithmetische Operationen auf Folgen

(a_n) und (b_n) seien konvergente Folgen mit den jeweiligen Grenzwerten a und b. Dann gilt:

1. $\lim (a_n \pm b_n) = a \pm b$;
2. $\lim (a_n \cdot b_n) = a \cdot b$;

Arithmetische Operationen auf Folgen

(a_n) und (b_n) seien konvergente Folgen mit den jeweiligen Grenzwerten a und b. Dann gilt:

1. $\lim (a_n \pm b_n) = a \pm b$;
2. $\lim (a_n \cdot b_n) = a \cdot b$;
3. $\lim \frac{a_n}{b_n} = \frac{a}{b}$, falls $b \neq 0$ und $b_n \neq 0$ für alle n.

Arithmetische Operationen auf Folgen

(a_n) und (b_n) seien konvergente Folgen mit den jeweiligen Grenzwerten a und b. Dann gilt:

1. $\lim (a_n \pm b_n) = a \pm b$;
2. $\lim (a_n \cdot b_n) = a \cdot b$;
3. $\lim \frac{a_n}{b_n} = \frac{a}{b}$, falls $b \neq 0$ und $b_n \neq 0$ für alle n.

1. ist einfach. Warum?

Arithmetische Operationen auf Folgen

(a_n) und (b_n) seien konvergente Folgen mit den jeweiligen Grenzwerten a und b. Dann gilt:

1. $\lim (a_n \pm b_n) = a \pm b$;
2. $\lim (a_n \cdot b_n) = a \cdot b$;
3. $\lim \frac{a_n}{b_n} = \frac{a}{b}$, falls $b \neq 0$ und $b_n \neq 0$ für alle n.

1. ist einfach. Warum? Ein Beweis für 2.

- Die konvergenten Folgen (a_n) und (b_n) sind beschränkt: (Warum?) Es gibt $c \in \mathbb{R}$ mit $|a_n|, |b_n| \leq c$ für alle n.

Arithmetische Operationen auf Folgen

(a_n) und (b_n) seien konvergente Folgen mit den jeweiligen Grenzwerten a und b. Dann gilt:

1. $\lim (a_n \pm b_n) = a \pm b$;
2. $\lim (a_n \cdot b_n) = a \cdot b$;
3. $\lim \frac{a_n}{b_n} = \frac{a}{b}$, falls $b \neq 0$ und $b_n \neq 0$ für alle n.

1. ist einfach. Warum? Ein Beweis für 2.

- Die konvergenten Folgen (a_n) und (b_n) sind beschränkt: (Warum?) Es gibt $c \in \mathbb{R}$ mit $|a_n|, |b_n| \leq c$ für alle n.
- Wie groß ist der Abstand zwischen $a_n b_n$ und ab?

$$|a_n b_n - ab| \quad = \quad |a_n b_n - a_n b + a_n b - ab|$$

Arithmetische Operationen auf Folgen

(a_n) und (b_n) seien konvergente Folgen mit den jeweiligen Grenzwerten a und b. Dann gilt:

1. $\lim (a_n \pm b_n) = a \pm b$;
2. $\lim (a_n \cdot b_n) = a \cdot b$;
3. $\lim \frac{a_n}{b_n} = \frac{a}{b}$, falls $b \neq 0$ und $b_n \neq 0$ für alle n.

1\. ist einfach. Warum? Ein Beweis für 2.

- Die konvergenten Folgen (a_n) und (b_n) sind beschränkt: (Warum?) Es gibt $c \in \mathbb{R}$ mit $|a_n|, |b_n| \leq c$ für alle n.
- Wie groß ist der Abstand zwischen $a_n b_n$ und ab?

$$\begin{aligned} |a_n b_n - ab| &= |a_n b_n - a_n b + a_n b - ab| \\ &\leq |a_n| \cdot |b_n - b| + |b| \cdot |a_n - a| \end{aligned}$$

Arithmetische Operationen auf Folgen

(a_n) und (b_n) seien konvergente Folgen mit den jeweiligen Grenzwerten a und b. Dann gilt:

1. $\lim\,(a_n \pm b_n) = a \pm b$;
2. $\lim\,(a_n \cdot b_n) = a \cdot b$;
3. $\lim \frac{a_n}{b_n} = \frac{a}{b}$, falls $b \neq 0$ und $b_n \neq 0$ für alle n.

1. ist einfach. Warum? Ein Beweis für 2.

- Die konvergenten Folgen (a_n) und (b_n) sind beschränkt: (Warum?) Es gibt $c \in \mathbb{R}$ mit $|a_n|, |b_n| \leq c$ für alle n.
- Wie groß ist der Abstand zwischen a_nb_n und ab?

$$
\begin{aligned}
|a_nb_n - ab| &= |a_nb_n - a_nb + a_nb - ab| \\
&\leq |a_n| \cdot |b_n - b| + |b| \cdot |a_n - a| \\
&\leq c \cdot \left(|b_n - b| + |a_n - a|\right).
\end{aligned}
$$

Stetige Funktionen

- Eine Funktion $f : [a, b] \to \mathbb{R}$ ist genau dann **stetig** im Punkt $c \in [a, b]$, wenn

$$\lim f(c_n) = f(c)$$

für jede Folge (c_n) mit $a \leq c_n \leq b$ und $\lim c_n = c$ gilt.

Stetige Funktionen

- Eine Funktion $f : [a, b] \to \mathbb{R}$ ist genau dann **stetig** im Punkt $c \in [a, b]$, wenn
$$\lim f(c_n) = f(c)$$
für jede Folge (c_n) mit $a \leq c_n \leq b$ und $\lim c_n = c$ gilt.
- Die Funktion f ist stetig, wenn f für alle Punkte $c \in [a, b]$ stetig ist.

Stetige Funktionen

- Eine Funktion $f : [a, b] \to \mathbb{R}$ ist genau dann **stetig** im Punkt $c \in [a, b]$, wenn

$$\lim f(c_n) = f(c)$$

  für jede Folge (c_n) mit $a \leq c_n \leq b$ und $\lim c_n = c$ gilt.
- Die Funktion f ist stetig, wenn f für alle Punkte $c \in [a, b]$ stetig ist.

- Der Graph einer stetigen Funktion macht „keine Sprünge“ und kann deshalb in einem Strich gezeichnet werden.

Stetige Funktionen

- Eine Funktion $f : [a, b] \to \mathbb{R}$ ist genau dann **stetig** im Punkt $c \in [a, b]$, wenn
$$\lim f(c_n) = f(c)$$
für jede Folge (c_n) mit $a \leq c_n \leq b$ und $\lim c_n = c$ gilt.
- Die Funktion f ist stetig, wenn f für alle Punkte $c \in [a, b]$ stetig ist.

- Der Graph einer stetigen Funktion macht „keine Sprünge“ und kann deshalb in einem Strich gezeichnet werden.
- Für eine stetige Funktion dürfen wir Funktion und Grenzwert vertauschen:

  Die Funktion $f : [a, b] \to \mathbb{R}$ sei stetig. Wenn alle Folgenglieder der Folge (c_n) zum Intervall $[a, b]$ gehören, dann folgt
$$\lim f(c_n) = f(\lim c_n).$$

Beispiele stetiger Funktionen

Wenn $f, g : [a, b] \to \mathbb{R}$ stetige Funktionen sind, dann sind auch

1. $f + g$ und $f \cdot g$ stetig.

Beispiele stetiger Funktionen

Wenn $f, g : [a, b] \to \mathbb{R}$ stetige Funktionen sind, dann sind auch

1. $f + g$ und $f \cdot g$ stetig.
2. Besitzt die Funktion g keine Nullstelle im Intervall $[a, b]$, dann ist auch f/g eine stetige Funktion.

Beispiele stetiger Funktionen

Wenn $f, g : [a, b] \to \mathbb{R}$ stetige Funktionen sind, dann sind auch

1. $f + g$ und $f \cdot g$ stetig.
2. Besitzt die Funktion g keine Nullstelle im Intervall $[a, b]$, dann ist auch f/g eine stetige Funktion.

- Warum stimmen die Behauptungen? Arithmetische Operationen auf konvergenten Folgen führen zu konvergenten Folgen.

Beispiele stetiger Funktionen

Wenn $f, g : [a, b] \to \mathbb{R}$ stetige Funktionen sind, dann sind auch

1. $f + g$ und $f \cdot g$ stetig.
2. Besitzt die Funktion g keine Nullstelle im Intervall $[a, b]$, dann ist auch f/g eine stetige Funktion.

- Warum stimmen die Behauptungen? Arithmetische Operationen auf konvergenten Folgen führen zu konvergenten Folgen.
- Aus 1. folgt, dass alle Polynome $p(x) = \sum_{k=0}^{n} a_k x^k$ stetig sind.

Beispiele stetiger Funktionen

Wenn $f, g : [a, b] \to \mathbb{R}$ stetige Funktionen sind, dann sind auch

1. $f + g$ und $f \cdot g$ stetig.
2. Besitzt die Funktion g keine Nullstelle im Intervall $[a, b]$, dann ist auch f/g eine stetige Funktion.

- Warum stimmen die Behauptungen? Arithmetische Operationen auf konvergenten Folgen führen zu konvergenten Folgen.
- Aus 1. folgt, dass alle Polynome $p(x) = \sum_{k=0}^{n} a_k x^k$ stetig sind.
- Als Konsequenz von 2. sind alle rationale Funktionen, also die Quotienten $p_1(x)/p_2(x)$ von Polynomen, stetig in allen Punkten, die keine Nullstelle des Nenners sind.

Folgen $a_{n+1} = f(a_n)$

- Angenommen, $a = \lim a_n$ existiert und wir sollen a bestimmen.
- Es gelte $a_{n+1} = f(a_n)$ für eine stetige Funktion f.

Folgen $a_{n+1} = f(a_n)$

- Angenommen, $a = \lim a_n$ existiert und wir sollen a bestimmen.
- Es gelte $a_{n+1} = f(a_n)$ für eine stetige Funktion f.

- Dann gilt

$$a = \lim_{n\to\infty} a_n$$

Folgen $a_{n+1} = f(a_n)$

- Angenommen, $a = \lim a_n$ existiert und wir sollen a bestimmen.
- Es gelte $a_{n+1} = f(a_n)$ für eine stetige Funktion f.

- Dann gilt

$$a = \lim_{n\to\infty} a_n = \lim_{n\to\infty} a_{n+1} =$$

Folgen $a_{n+1} = f(a_n)$

- Angenommen, $a = \lim a_n$ existiert und wir sollen a bestimmen.
- Es gelte $a_{n+1} = f(a_n)$ für eine stetige Funktion f.

- Dann gilt

$$a = \lim_{n\to\infty} a_n = \lim_{n\to\infty} a_{n+1} = \lim_{n\to\infty} f(a_n)$$

Folgen $a_{n+1} = f(a_n)$

- Angenommen, $a = \lim a_n$ existiert und wir sollen a bestimmen.
- Es gelte $a_{n+1} = f(a_n)$ für eine stetige Funktion f.

- Dann gilt

$$a = \lim_{n\to\infty} a_n = \lim_{n\to\infty} a_{n+1} = \lim_{n\to\infty} f(a_n) = f(a).$$

Folgen $a_{n+1} = f(a_n)$

- Angenommen, $a = \lim a_n$ existiert und wir sollen a bestimmen.
- Es gelte $a_{n+1} = f(a_n)$ für eine stetige Funktion f.

- Dann gilt

$$a = \lim_{n\to\infty} a_n = \lim_{n\to\infty} a_{n+1} = \lim_{n\to\infty} f(a_n) = f(a).$$

- Der Grenzwert a erfüllt also die Beziehung $f(a) = a$, aus der wir hoffentlich den gesuchten Grenzwert a berechnen können.

Die Berechnung der Quadratwurzel $\sqrt{b}$

Für $b \in \mathbb{R}$, $b > 0$ betrachten wir die Folge (a_n) mit $a_0 = b + 1$ und

$$a_{n+1} = \frac{1}{2}\left(a_n + \frac{b}{a_n}\right).$$

Die Berechnung der Quadratwurzel $\sqrt{b}$

Für $b \in \mathbb{R}$, $b > 0$ betrachten wir die Folge (a_n) mit $a_0 = b + 1$ und

$$a_{n+1} = \frac{1}{2}\left(a_n + \frac{b}{a_n}\right).$$

- Angenommen wir wissen, dass (a_n) konvergiert.
- Wir wenden die Grenzwertregeln auf

$$a_{n+1} = f(a_n) \ \text{ mit } f(x) = \frac{1}{2}\left(x + \frac{b}{x}\right)$$

  an und erhalten die Gleichung

Die Berechnung der Quadratwurzel $\sqrt{b}$

Für $b \in \mathbb{R}$, $b > 0$ betrachten wir die Folge (a_n) mit $a_0 = b + 1$ und

$$a_{n+1} = \frac{1}{2}\left(a_n + \frac{b}{a_n}\right).$$

- Angenommen wir wissen, dass (a_n) konvergiert.
- Wir wenden die Grenzwertregeln auf

$$a_{n+1} = f(a_n) \text{ mit } f(x) = \frac{1}{2}\left(x + \frac{b}{x}\right)$$

  an und erhalten die Gleichung
  - $a = f(a) = \frac{1}{2}\left(a + \frac{b}{a}\right)$.

Die Berechnung der Quadratwurzel $\sqrt{b}$

Für $b \in \mathbb{R}$, $b > 0$ betrachten wir die Folge (a_n) mit $a_0 = b + 1$ und

$$a_{n+1} = \frac{1}{2}\left(a_n + \frac{b}{a_n}\right).$$

- Angenommen wir wissen, dass (a_n) konvergiert.
- Wir wenden die Grenzwertregeln auf

$$a_{n+1} = f(a_n) \text{ mit } f(x) = \frac{1}{2}\left(x + \frac{b}{x}\right)$$

  an und erhalten die Gleichung
  - $a = f(a) = \frac{1}{2}\left(a + \frac{b}{a}\right)$.
  - $\lim a_n = a = \sqrt{b}$ folgt.

Die Berechnung der Quadratwurzel $\sqrt{b}$

Für $b \in \mathbb{R}$, $b > 0$ betrachten wir die Folge (a_n) mit $a_0 = b + 1$ und

$$a_{n+1} = \frac{1}{2}\left(a_n + \frac{b}{a_n}\right).$$

- Angenommen wir wissen, dass (a_n) konvergiert.
- Wir wenden die Grenzwertregeln auf

$$a_{n+1} = f(a_n) \;\text{ mit } f(x) = \frac{1}{2}\left(x + \frac{b}{x}\right)$$

  an und erhalten die Gleichung
    - $a = f(a) = \frac{1}{2}\left(a + \frac{b}{a}\right)$.
    - $\lim a_n = a = \sqrt{b}$ folgt.
- Warum konvergiert (a_n)?

Das Monotoniekriterium

- (a_n) sei eine monoton fallende oder monoton wachsende Folge.
- Wenn die Folge (a_n) beschränkt ist, dann konvergiert (a_n).

Das Monotoniekriterium

- (a_n) sei eine monoton fallende oder monoton wachsende Folge.
- Wenn die Folge (a_n) beschränkt ist, dann konvergiert (a_n).

- Warum ist die Behauptung richtig?

Das Monotoniekriterium

- (a_n) sei eine monoton fallende oder monoton wachsende Folge.
- Wenn die Folge (a_n) beschränkt ist, dann konvergiert (a_n).

- Warum ist die Behauptung richtig?

  Die Behauptung kann nicht gezeigt werden, sondern muss als Axiom gefordert werden.

Das Monotoniekriterium

- (a_n) sei eine monoton fallende oder monoton wachsende Folge.
- Wenn die Folge (a_n) beschränkt ist, dann konvergiert (a_n).

- Warum ist die Behauptung richtig?

  Die Behauptung kann nicht gezeigt werden, sondern muss als Axiom gefordert werden.
- Es ist $a_{n+1} = \frac{1}{2}\left(a_n + \frac{b}{a_n}\right)$. Die Folge konvergiert, weil:
  - $a_n > 0$ für alle n aus $a_0 > 1$ und $b > 0$ folgt.

Das Monotoniekriterium

- (a_n) sei eine monoton fallende oder monoton wachsende Folge.
- Wenn die Folge (a_n) beschränkt ist, dann konvergiert (a_n).

- Warum ist die Behauptung richtig?

  Die Behauptung kann nicht gezeigt werden, sondern muss als Axiom gefordert werden.
- Es ist $a_{n+1} = \frac{1}{2}\left(a_n + \frac{b}{a_n}\right)$. Die Folge konvergiert, weil:
  - $a_n > 0$ für alle n aus $a_0 > 1$ und $b > 0$ folgt.
  - Zeige: Die Folge ist monoton fallend.

Dividieren mit Additionen und Muliplikationen

Sei $a > 1$ eine reelle Zahl.
Wir wollen $1/a$ berechnen, ohne Divisionen zu benutzen.

Dividieren mit Additionen und Muliplikationen

Sei $a > 1$ eine reelle Zahl.
Wir wollen $1/a$ berechnen, ohne Divisionen zu benutzen.

- Wir suchen eine Lösung x für die Gleichung $ax = 1$, bzw. für $x = 2x - ax^2$.

Dividieren mit Additionen und Muliplikationen

Sei $a > 1$ eine reelle Zahl.
Wir wollen $1/a$ berechnen, ohne Divisionen zu benutzen.

- Wir suchen eine Lösung x für die Gleichung $ax = 1$,
  bzw. für $x = 2x - ax^2$.

    Das gesuchte x ist dann die von Null verschiedene Lösung dieser Gleichung.

Dividieren mit Additionen und Muliplikationen

Sei $a > 1$ eine reelle Zahl.
Wir wollen $1/a$ berechnen, ohne Divisionen zu benutzen.

- Wir suchen eine Lösung x für die Gleichung $ax = 1$, bzw. für $x = 2x - ax^2$.

  Das gesuchte x ist dann die von Null verschiedene Lösung dieser Gleichung.
- Für $f(x) := 2x - ax^2$ erhalten wir die Rekursionsgleichung

Dividieren mit Additionen und Muliplikationen

Sei $a > 1$ eine reelle Zahl.
Wir wollen $1/a$ berechnen, ohne Divisionen zu benutzen.

- Wir suchen eine Lösung x für die Gleichung $ax = 1$, bzw. für $x = 2x - ax^2$.

  Das gesuchte x ist dann die von Null verschiedene Lösung dieser Gleichung.

- Für $f(x) := 2x - ax^2$ erhalten wir die Rekursionsgleichung

$$x_{n+1} = 2x_n - ax_n^2.$$

Dividieren mit Additionen und Muliplikationen

Sei $a > 1$ eine reelle Zahl.
Wir wollen $1/a$ berechnen, ohne Divisionen zu benutzen.

- Wir suchen eine Lösung x für die Gleichung $ax = 1$, bzw. für $x = 2x - ax^2$.

  Das gesuchte x ist dann die von Null verschiedene Lösung dieser Gleichung.

- Für $f(x) := 2x - ax^2$ erhalten wir die Rekursionsgleichung

$$x_{n+1} = 2x_n - ax_n^2.$$

- Zeige, dass die Folge (x_n) mit $x_0 \geq 1/a$ gegen $1/a$ strebt.

Die Regeln von Bernoulli–l'Hospital

- f und g seien differenzierbare Funktionen.
- Die Folge (a_n) konvergiere gegen $a \in \mathbb{R} \cup \{-\infty, \infty\}$.
- Wenn $\lim f(a_n) = \lim g(a_n) = 0$ oder $\lim |f(a_n)| = \lim |g(a_n)| = \infty$, dann folgt

$$\lim \frac{f(a_n)}{g(a_n)} = \lim \frac{f'(a_n)}{g'(a_n)} ,$$

falls der zweite Grenzwert existiert.

Die Regeln von Bernoulli–l'Hospital

- f und g seien differenzierbare Funktionen.
- Die Folge (a_n) konvergiere gegen $a \in \mathbb{R} \cup \{-\infty, \infty\}$.
- Wenn $\lim f(a_n) = \lim g(a_n) = 0$ oder $\lim |f(a_n)| = \lim |g(a_n)| = \infty$, dann folgt

$$\lim \frac{f(a_n)}{g(a_n)} = \lim \frac{f'(a_n)}{g'(a_n)},$$

falls der zweite Grenzwert existiert.

Zum Beispiel $\lim \frac{1}{n} = \lim \frac{0}{1} = 0$: Wähle $f(x) = 1$ und $g(x) = x$.

Die Regeln von Bernoulli–l'Hospital

- f und g seien differenzierbare Funktionen.
- Die Folge (a_n) konvergiere gegen $a \in \mathbb{R} \cup \{-\infty, \infty\}$.
- Wenn $\lim f(a_n) = \lim g(a_n) = 0$ oder $\lim |f(a_n)| = \lim |g(a_n)| = \infty$, dann folgt

$$\lim \frac{f(a_n)}{g(a_n)} = \lim \frac{f'(a_n)}{g'(a_n)} ,$$

falls der zweite Grenzwert existiert.

Zum Beispiel $\lim \frac{1}{n} = \lim \frac{0}{1} = 0$: Wähle $f(x) = 1$ und $g(x) = x$.

Wie differenziert man?

Ableitungen häufig benutzter Funktionen

Für beliebige Konstanten $c \in \mathbb{R}$ und $n \in \mathbb{N}$:

1. $c' = 0$;

Ableitungen häufig benutzter Funktionen

Für beliebige Konstanten $c \in \mathbb{R}$ und $n \in \mathbb{N}$:

1. $c' = 0$;
2. $(x^n)' = nx^{n-1}$;

Ableitungen häufig benutzter Funktionen

Für beliebige Konstanten $c \in \mathbb{R}$ und $n \in \mathbb{N}$:

1. $c' = 0$;
2. $(x^n)' = nx^{n-1}$;
3. $(1/x)' = -1/x^2$;

Ableitungen häufig benutzter Funktionen

Für beliebige Konstanten $c \in \mathbb{R}$ und $n \in \mathbb{N}$:

1. $c' = 0$;
2. $(x^n)' = nx^{n-1}$;
3. $(1/x)' = -1/x^2$;
4. $(e^x)' = e^x$;

Ableitungen häufig benutzter Funktionen

Für beliebige Konstanten $c \in \mathbb{R}$ und $n \in \mathbb{N}$:

1. $c' = 0$;
2. $(x^n)' = nx^{n-1}$;
3. $(1/x)' = -1/x^2$;
4. $(e^x)' = e^x$;
5. $(\ln x)' = 1/x$.

Ableitungsregeln

1. **Summenregel**: $(f \pm g)' = f' \pm g'$.

Ableitungsregeln

1. **Summenregel**: $(f \pm g)' = f' \pm g'$.
2. **Produktregel**: $(f \cdot g)' = f' \cdot g + f \cdot g'$.

Ableitungsregeln

1. **Summenregel**: $(f \pm g)' = f' \pm g'$.
2. **Produktregel**: $(f \cdot g)' = f' \cdot g + f \cdot g'$.
3. **Quotientenregel**: $\left(\frac{f}{g}\right)' = \frac{f' \cdot g - f \cdot g'}{g^2}$.

Ableitungsregeln

1. **Summenregel**: $(f \pm g)' = f' \pm g'$.
2. **Produktregel**: $(f \cdot g)' = f' \cdot g + f \cdot g'$.
3. **Quotientenregel**: $\left(\frac{f}{g}\right)' = \frac{f' \cdot g - f \cdot g'}{g^2}$.
4. **Kettenregel**: Ist $\varphi(x) = f(g(x))$, so gilt $\varphi'(x) = f'(g(x)) \cdot g'(x)$.

Wer wächst schneller: n^k oder e^n?

- Wir wählen $k = 2$:

Wer wächst schneller: n^k oder e^n?

- Wir wählen $k = 2$:
  - Es ist $\lim n^2 = \lim e^n = \infty$.

Wer wächst schneller: n^k oder e^n?

- Wir wählen $k = 2$:
  - Es ist $\lim n^2 = \lim e^n = \infty$.
  - Wende die Regel von de l'Hospital an:

$$\lim \frac{n^2}{e^n} = \lim \frac{2 \cdot n}{e^n}$$

Wer wächst schneller: n^k oder e^n?

- Wir wählen $k = 2$:
  - Es ist $\lim n^2 = \lim e^n = \infty$.
  - Wende die Regel von de l'Hospital an:

$$\lim \frac{n^2}{e^n} = \lim \frac{2 \cdot n}{e^n} = \lim \frac{2}{e^n}$$

Wer wächst schneller: n^k oder e^n?

- Wir wählen $k = 2$:
  - Es ist $\lim n^2 = \lim e^n = \infty$.
  - Wende die Regel von de l'Hospital an:

$$\lim \frac{n^2}{e^n} = \lim \frac{2 \cdot n}{e^n} = \lim \frac{2}{e^n} = 0\,.$$

Wer wächst schneller: n^k oder e^n?

- Wir wählen $k = 2$:
  - Es ist $\lim n^2 = \lim e^n = \infty$.
  - Wende die Regel von de l'Hospital an:

$$\lim \frac{n^2}{e^n} = \lim \frac{2 \cdot n}{e^n} = \lim \frac{2}{e^n} = 0\,.$$

- Und für $k = 3, 4, \ldots$?

Wer wächst schneller: n^k oder e^n?

- Wir wählen $k = 2$:
  - Es ist $\lim n^2 = \lim e^n = \infty$.
  - Wende die Regel von de l'Hospital an:

$$\lim \frac{n^2}{e^n} = \lim \frac{2 \cdot n}{e^n} = \lim \frac{2}{e^n} = 0\,.$$

- Und für $k = 3, 4, \ldots$?

- Die Exponentialfunktion e^n wächst rasant schneller als jede Potenz n^k.

Wer wächst schneller: n^k oder e^n?

- Wir wählen $k = 2$:
  - Es ist $\lim n^2 = \lim e^n = \infty$.
  - Wende die Regel von de l'Hospital an:

$$\lim \frac{n^2}{e^n} = \lim \frac{2 \cdot n}{e^n} = \lim \frac{2}{e^n} = 0\,.$$

- Und für $k = 3, 4, \ldots$?

- Die Exponentialfunktion e^n wächst rasant schneller als jede Potenz n^k.
- Ein Algorithmus polynomieller Laufzeit,

  also mit Laufzeit höchstens $c \cdot n^k$ für Konstanten c und k,

  ist einem Algorithmus mit exponentieller Laufzeit vorzuziehen.

Die asymptotische Notation

$f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ seien Funktionen,

Die asymptotische Notation

$f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ seien Funktionen, die einer Eingabelänge $n \in \mathbb{N}$ eine nicht-negative Laufzeit $f(n)$, bzw. $g(n)$ zuweisen.

Die asymptotische Notation

$f, g : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ seien Funktionen, die einer Eingabelänge $n \in \mathbb{N}$ eine nicht-negative Laufzeit $f(n)$, bzw. $g(n)$ zuweisen.

- Die Groß-Oh Notation: $f = O(g)$ $\Leftrightarrow$ Es gibt eine positive Konstante $c > 0$ und eine natürliche Zahl $n_0 \in \mathbb{N}$,

$f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ seien Funktionen, die einer Eingabelänge $n \in \mathbb{N}$ eine nicht-negative Laufzeit $f(n)$, bzw. $g(n)$ zuweisen.

- Die Groß-Oh Notation: $f = O(g) \Leftrightarrow$ Es gibt eine positive Konstante $c > 0$ und eine natürliche Zahl $n_0 \in \mathbb{N}$, so dass

  $$f(n) \leq c \cdot g(n)$$

  für alle $n \geq n_0$ gilt:

Die asymptotische Notation

$f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ seien Funktionen, die einer Eingabelänge $n \in \mathbb{N}$ eine nicht-negative Laufzeit $f(n)$, bzw. $g(n)$ zuweisen.

- Die Groß-Oh Notation: $f = O(g) \Leftrightarrow$ Es gibt eine positive Konstante $c > 0$ und eine natürliche Zahl $n_0 \in \mathbb{N}$, so dass

  $$f(n) \leq c \cdot g(n)$$

  für alle $n \geq n_0$ gilt: f wächst höchstens so schnell wie g.

Die asymptotische Notation

$f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ seien Funktionen, die einer Eingabelänge $n \in \mathbb{N}$ eine nicht-negative Laufzeit $f(n)$, bzw. $g(n)$ zuweisen.

- Die Groß-Oh Notation: $f = O(g) \Leftrightarrow$ Es gibt eine positive Konstante $c > 0$ und eine natürliche Zahl $n_0 \in \mathbb{N}$, so dass

  $$f(n) \leq c \cdot g(n)$$

  für alle $n \geq n_0$ gilt: f wächst höchstens so schnell wie g.
- $f = \Theta(g) \Leftrightarrow f = O(g)$ und $g = O(f)$:

Die asymptotische Notation

$f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ seien Funktionen, die einer Eingabelänge $n \in \mathbb{N}$ eine nicht-negative Laufzeit $f(n)$, bzw. $g(n)$ zuweisen.

- Die Groß-Oh Notation: $f = O(g) \Leftrightarrow$ Es gibt eine positive Konstante $c > 0$ und eine natürliche Zahl $n_0 \in \mathbb{N}$, so dass

  $$f(n) \leq c \cdot g(n)$$

  für alle $n \geq n_0$ gilt: f wächst höchstens so schnell wie g.
- $f = \Theta(g) \Leftrightarrow f = O(g)$ und $g = O(f)$:
  f und g wachsen gleich schnell.

Die asymptotische Notation

$f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ seien Funktionen, die einer Eingabelänge $n \in \mathbb{N}$ eine nicht-negative Laufzeit $f(n)$, bzw. $g(n)$ zuweisen.

- Die Groß-Oh Notation: $f = O(g) \Leftrightarrow$ Es gibt eine positive Konstante $c > 0$ und eine natürliche Zahl $n_0 \in \mathbb{N}$, so dass

$$f(n) \leq c \cdot g(n)$$

  für alle $n \geq n_0$ gilt: f wächst höchstens so schnell wie g.
- $f = \Theta(g) \Leftrightarrow f = O(g)$ und $g = O(f)$:
  f und g wachsen gleich schnell.
- Die Klein-Oh Notation: $f = o(g) \Leftrightarrow \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$:

Die asymptotische Notation

$f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$ seien Funktionen, die einer Eingabelänge $n \in \mathbb{N}$ eine nicht-negative Laufzeit $f(n)$, bzw. $g(n)$ zuweisen.

- Die Groß-Oh Notation: $f = O(g) \Leftrightarrow$ Es gibt eine positive Konstante $c > 0$ und eine natürliche Zahl $n_0 \in \mathbb{N}$, so dass
$$f(n) \leq c \cdot g(n)$$
für alle $n \geq n_0$ gilt: f wächst höchstens so schnell wie g.
- $f = \Theta(g) \Leftrightarrow f = O(g)$ und $g = O(f)$:
f und g wachsen gleich schnell.
- Die Klein-Oh Notation: $f = o(g) \Leftrightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$:
f wächst langsamer als g.

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n \to \infty} \frac{f(n)}{g(n)} = c$.

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n \to \infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$.

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n \to \infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)
- Wenn $0 < c < \infty$, dann ist

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)
- Wenn $0 < c < \infty$, dann ist $f = \Theta(g)$.

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)
- Wenn $0 < c < \infty$, dann ist $f = \Theta(g)$.
  (f und g wachsen gleich schnell.)

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)
- Wenn $0 < c < \infty$, dann ist $f = \Theta(g)$.
  (f und g wachsen gleich schnell.)
- Wenn $c = \infty$, dann ist

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n \to \infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)
- Wenn $0 < c < \infty$, dann ist $f = \Theta(g)$.
  (f und g wachsen gleich schnell.)
- Wenn $c = \infty$, dann ist $g = o(f)$.

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)
- Wenn $0 < c < \infty$, dann ist $f = \Theta(g)$.
  (f und g wachsen gleich schnell.)
- Wenn $c = \infty$, dann ist $g = o(f)$. (f wächst schneller als g.)

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)
- Wenn $0 < c < \infty$, dann ist $f = \Theta(g)$.
  (f und g wachsen gleich schnell.)
- Wenn $c = \infty$, dann ist $g = o(f)$. (f wächst schneller als g.)
- Wenn $0 \leq c < \infty$, dann ist

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)
- Wenn $0 < c < \infty$, dann ist $f = \Theta(g)$.
  (f und g wachsen gleich schnell.)
- Wenn $c = \infty$, dann ist $g = o(f)$. (f wächst schneller als g.)
- Wenn $0 \leq c < \infty$, dann ist $f = O(g)$.

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)
- Wenn $0 < c < \infty$, dann ist $f = \Theta(g)$.
  (f und g wachsen gleich schnell.)
- Wenn $c = \infty$, dann ist $g = o(f)$. (f wächst schneller als g.)
- Wenn $0 \leq c < \infty$, dann ist $f = O(g)$.
  (f wächst höchstens so schnell wie g.)

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n \to \infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)
- Wenn $0 < c < \infty$, dann ist $f = \Theta(g)$.
  (f und g wachsen gleich schnell.)
- Wenn $c = \infty$, dann ist $g = o(f)$. (f wächst schneller als g.)
- Wenn $0 \leq c < \infty$, dann ist $f = O(g)$.
  (f wächst höchstens so schnell wie g.)
- Wenn $0 < c \leq \infty$, dann ist

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)
- Wenn $0 < c < \infty$, dann ist $f = \Theta(g)$.
  (f und g wachsen gleich schnell.)
- Wenn $c = \infty$, dann ist $g = o(f)$. (f wächst schneller als g.)
- Wenn $0 \leq c < \infty$, dann ist $f = O(g)$.
  (f wächst höchstens so schnell wie g.)
- Wenn $0 < c \leq \infty$, dann ist $g = O(f)$.

Asymptotische Notation und Grenzwerte

Grenzwerte sollten das Wachstum voraussagen!

Der Grenzwert der Folge $\frac{f(n)}{g(n)}$ existiere und es sei $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$.

- Wenn $c = 0$, dann ist $f = o(g)$. (f wächst langsamer als g.)
- Wenn $0 < c < \infty$, dann ist $f = \Theta(g)$.
  (f und g wachsen gleich schnell.)
- Wenn $c = \infty$, dann ist $g = o(f)$. (f wächst schneller als g.)
- Wenn $0 \leq c < \infty$, dann ist $f = O(g)$.
  (f wächst höchstens so schnell wie g.)
- Wenn $0 < c \leq \infty$, dann ist $g = O(f)$.
  (f wächst mindestens so schnell wie g.)

Eine Wachstums-Hierarchie

- $1 = o(\log_2 n)$,

Eine Wachstums-Hierarchie

- $1 = o(\log_2 n)$,
  $\lim_{n\to\infty} \log_2 n = \infty$.

Eine Wachstums-Hierarchie

- $1 = o(\log_2 n)$,
  $\lim_{n\to\infty} \log_2 n = \infty$.
- $(\log_2 n)^k = o(n)$ für jedes k,

Eine Wachstums-Hierarchie

- $1 = o(\log_2 n)$,
  $\lim_{n\to\infty} \log_2 n = \infty$.
- $(\log_2 n)^k = o(n)$ für jedes k,
  - Für $k = 2$ mit de l'Hospital:
    $\lim \frac{(\log_2 n)^2}{n} = \lim \frac{(2 \log_2 n)/n}{1}$

Eine Wachstums-Hierarchie

- $1 = o(\log_2 n)$,
  $\lim_{n\to\infty} \log_2 n = \infty$.
- $(\log_2 n)^k = o(n)$ für jedes k,
  - Für $k = 2$ mit de l'Hospital:
    $\lim \frac{(\log_2 n)^2}{n} = \lim \frac{(2\log_2 n)/n}{1} = \lim \frac{2\log_2 n}{n}$

Eine Wachstums-Hierarchie

- $1 = o(\log_2 n)$,

  $\lim_{n\to\infty} \log_2 n = \infty$.

- $(\log_2 n)^k = o(n)$ für jedes k,
  - Für $k = 2$ mit de l'Hospital:

    $\lim \frac{(\log_2 n)^2}{n} = \lim \frac{(2\log_2 n)/n}{1} = \lim \frac{2\log_2 n}{n} = \lim \frac{2}{n}$

Eine Wachstums-Hierarchie

- $1 = o(\log_2 n)$,
  $\lim_{n\to\infty} \log_2 n = \infty$.
- $(\log_2 n)^k = o(n)$ für jedes k,
  - Für $k = 2$ mit de l'Hospital:
    $\lim \frac{(\log_2 n)^2}{n} = \lim \frac{(2\log_2 n)/n}{1} = \lim \frac{2\log_2 n}{n} = \lim \frac{2}{n} = 0.$
  - Binärsuche ist viel schneller als lineare Suche!

Eine Wachstums-Hierarchie

- $1 = o(\log_2 n)$,
  $\lim_{n\to\infty} \log_2 n = \infty$.
- $(\log_2 n)^k = o(n)$ für jedes k,
  - Für $k = 2$ mit de l'Hospital:
    $\lim \frac{(\log_2 n)^2}{n} = \lim \frac{(2\log_2 n)/n}{1} = \lim \frac{2\log_2 n}{n} = \lim \frac{2}{n} = 0.$
  - Binärsuche ist viel schneller als lineare Suche!
- $n = o(n \cdot \log_2 n)$,

Eine Wachstums-Hierarchie

- $1 = o(\log_2 n)$,
  $\lim_{n\to\infty} \log_2 n = \infty$.
- $(\log_2 n)^k = o(n)$ für jedes k,
  - Für $k = 2$ mit de l'Hospital:
    $\lim \frac{(\log_2 n)^2}{n} = \lim \frac{(2 \log_2 n)/n}{1} = \lim \frac{2 \log_2 n}{n} = \lim \frac{2}{n} = 0.$
  - Binärsuche ist viel schneller als lineare Suche!
- $n = o(n \cdot \log_2 n)$,
  $\lim_{n\to\infty} \frac{n}{n \log n} = \lim_{n\to\infty} \frac{1}{\log_2 n} = 0.$

Eine Wachstums-Hierarchie

- $1 = o(\log_2 n)$,

  $\lim_{n\to\infty} \log_2 n = \infty$.
- $(\log_2 n)^k = o(n)$ für jedes k,
  - Für $k = 2$ mit de l'Hospital:

    $\lim \frac{(\log_2 n)^2}{n} = \lim \frac{(2\log_2 n)/n}{1} = \lim \frac{2\log_2 n}{n} = \lim \frac{2}{n} = 0.$
  - Binärsuche ist viel schneller als lineare Suche!
- $n = o(n \cdot \log_2 n)$,

  $\lim_{n\to\infty} \frac{n}{n \log n} = \lim_{n\to\infty} \frac{1}{\log_2 n} = 0.$
- $n \cdot \log_2 n = o(n^2)$. Warum?

  Mergesort ist viel schneller als Bubble Sort.

Eine Wachstums-Hierarchie

- $1 = o(\log_2 n)$,

  $\lim_{n\to\infty} \log_2 n = \infty$.
- $(\log_2 n)^k = o(n)$ für jedes k,
  - Für $k = 2$ mit de l'Hospital:
    $\lim \frac{(\log_2 n)^2}{n} = \lim \frac{(2\log_2 n)/n}{1} = \lim \frac{2\log_2 n}{n} = \lim \frac{2}{n} = 0.$
  - Binärsuche ist viel schneller als lineare Suche!
- $n = o(n \cdot \log_2 n)$,

  $\lim_{n\to\infty} \frac{n}{n \log n} = \lim_{n\to\infty} \frac{1}{\log_2 n} = 0.$
- $n \cdot \log_2 n = o(n^2)$. Warum?

  Mergesort ist viel schneller als Bubble Sort.
- $n^k = o(e^n)$.

Logarithmisch, linear, quadratisch, ... , exponentiell, ...

$$f = O(1): \qquad f \text{ ist konstant, } f(n) \leq c$$
$$f = \Theta(\ln n): \qquad f \text{ wächst logarithmisch,}$$

Logarithmisch, linear, quadratisch, ... , exponentiell, ...

$f = O(1)$: f ist konstant, $f(n) \leq c$

$f = \Theta(\ln n)$: f wächst logarithmisch, $f(2n) \approx f(n) + c$

$f = \Theta(\sqrt{n})$: f wächst wie die Wurzelfunktion,

Logarithmisch, linear, quadratisch, ... , exponentiell, ...

$$
\begin{aligned}
f = O(1):&\quad f \text{ ist konstant, } f(n) \leq c \\
f = \Theta(\ln n):&\quad f \text{ wächst logarithmisch, } f(2n) \approx f(n) + c \\
f = \Theta(\sqrt{n}):&\quad f \text{ wächst wie die Wurzelfunktion, } f(4n) \approx 2 \cdot f(n) \\
f = \Theta(n):&\quad f \text{ hat lineares Wachstum,}
\end{aligned}
$$

Logarithmisch, linear, quadratisch, ... , exponentiell, ...

$$
\begin{aligned}
f = O(1):& \quad f \text{ ist konstant, } f(n) \leq c \\
f = \Theta(\ln n):& \quad f \text{ wächst logarithmisch, } f(2n) \approx f(n) + c \\
f = \Theta(\sqrt{n}):& \quad f \text{ wächst wie die Wurzelfunktion, } f(4n) \approx 2 \cdot f(n) \\
f = \Theta(n):& \quad f \text{ hat lineares Wachstum, } f(2n) \approx 2 \cdot f(n) \\
f = \Theta(n^2):& \quad f \text{ wächst quadratisch,}
\end{aligned}
$$

Logarithmisch, linear, quadratisch, ... , exponentiell, ...

$$
\begin{aligned}
f = O(1): &\quad f \text{ ist konstant, } f(n) \le c \\
f = \Theta(\ln n): &\quad f \text{ wächst logarithmisch, } f(2n) \approx f(n) + c \\
f = \Theta(\sqrt{n}): &\quad f \text{ wächst wie die Wurzelfunktion, } f(4n) \approx 2 \cdot f(n) \\
f = \Theta(n): &\quad f \text{ hat lineares Wachstum, } f(2n) \approx 2 \cdot f(n) \\
f = \Theta(n^2): &\quad f \text{ wächst quadratisch, } f(2n) \approx 4 \cdot f(n) \\
f = \Theta(2^n): &\quad f \text{ wächst exponentiell,}
\end{aligned}
$$

Logarithmisch, linear, quadratisch, ... , exponentiell, ...

$$
\begin{aligned}
f = O(1):& \quad f \text{ ist konstant, } f(n) \leq c \\
f = \Theta(\ln n):& \quad f \text{ wächst logarithmisch, } f(2n) \approx f(n) + c \\
f = \Theta(\sqrt{n}):& \quad f \text{ wächst wie die Wurzelfunktion, } f(4n) \approx 2 \cdot f(n) \\
f = \Theta(n):& \quad f \text{ hat lineares Wachstum, } f(2n) \approx 2 \cdot f(n) \\
f = \Theta(n^2):& \quad f \text{ wächst quadratisch, } f(2n) \approx 4 \cdot f(n) \\
f = \Theta(2^n):& \quad f \text{ wächst exponentiell, } f(n+1) \approx 2 \cdot f(n) \\
f = \Theta(n!):& \quad f \text{ hat faktorielles Wachstum,}
\end{aligned}
$$

Logarithmisch, linear, quadratisch, ... , exponentiell, ...

$$\begin{aligned}
f = O(1): &\quad f \text{ ist konstant, } f(n) \leq c \\
f = \Theta(\ln n): &\quad f \text{ wächst logarithmisch, } f(2n) \approx f(n) + c \\
f = \Theta(\sqrt{n}): &\quad f \text{ wächst wie die Wurzelfunktion, } f(4n) \approx 2 \cdot f(n) \\
f = \Theta(n): &\quad f \text{ hat lineares Wachstum, } f(2n) \approx 2 \cdot f(n) \\
f = \Theta(n^2): &\quad f \text{ wächst quadratisch, } f(2n) \approx 4 \cdot f(n) \\
f = \Theta(2^n): &\quad f \text{ wächst exponentiell, } f(n+1) \approx 2 \cdot f(n) \\
f = \Theta(n!): &\quad f \text{ hat faktorielles Wachstum, } f(n+1) \approx n \cdot f(n)
\end{aligned}$$

Unendliche Reihen

Was ist der Wert der unendlichen Reihe

$$S = \sum_{k=0}^{\infty} (-1)^k ?$$

Unendliche Reihen

Was ist der Wert der unendlichen Reihe

$$S = \sum_{k=0}^{\infty} (-1)^k ?$$

- $S = \underbrace{(1-1)}_{0} + \underbrace{(1-1)}_{0} + \underbrace{(1-1)}_{0} + \cdots = 0$

Unendliche Reihen

Was ist der Wert der unendlichen Reihe

$$S = \sum_{k=0}^{\infty} (-1)^k ?$$

- $S = \underbrace{(1-1)}_{0} + \underbrace{(1-1)}_{0} + \underbrace{(1-1)}_{0} + \cdots = 0$ oder
- $S = 1 + \underbrace{(-1+1)}_{0} + \underbrace{(-1+1)}_{0} + \underbrace{(-1+1)}_{0} + \cdots = 1?$

Unendliche Reihen

Was ist der Wert der unendlichen Reihe

$$S = \sum_{k=0}^{\infty} (-1)^k ?$$

- $S = \underbrace{(1-1)}_{0} + \underbrace{(1-1)}_{0} + \underbrace{(1-1)}_{0} + \cdots = 0$ oder
- $S = 1 + \underbrace{(-1+1)}_{0} + \underbrace{(-1+1)}_{0} + \underbrace{(-1+1)}_{0} + \cdots = 1?$

Wie sollten wir die Konvergenz einer unendlichen Reihe definieren?

Konvergenz unendlicher Reihen

Wir sagen, dass die unendliche Summe $\sum_{k=0}^{\infty} a_k$ gegen S konvergiert, wenn die Folge (S_n) mit $S_n = \sum_{k=0}^{n} a_k$ gegen S konvergiert.

Konvergenz unendlicher Reihen

Wir sagen, dass die unendliche Summe $\sum_{k=0}^{\infty} a_k$ gegen S konvergiert, wenn die Folge (S_n) mit $S_n = \sum_{k=0}^{n} a_k$ gegen S konvergiert.

- Die Reihe $\sum_k (-1)^k$ ist divergent, denn $1, 0, 1, 0 \ldots, 1, 0, \ldots$ ist die Folge der Partialsummen.

Konvergenz unendlicher Reihen

Wir sagen, dass die unendliche Summe $\sum_{k=0}^{\infty} a_k$ gegen S konvergiert, wenn die Folge (S_n) mit $S_n = \sum_{k=0}^{n} a_k$ gegen S konvergiert.

- Die Reihe $\sum_k (-1)^k$ ist divergent, denn $1, 0, 1, 0 \ldots, 1, 0, \ldots$ ist die Folge der Partialsummen.
- Zwei wichtige Reihen:
  - $\sum_k \frac{1}{k} =$

Konvergenz unendlicher Reihen

Wir sagen, dass die unendliche Summe $\sum_{k=0}^{\infty} a_k$ gegen S konvergiert, wenn die Folge (S_n) mit $S_n = \sum_{k=0}^{n} a_k$ gegen S konvergiert.

- Die Reihe $\sum_k (-1)^k$ ist divergent, denn $1, 0, 1, 0 \ldots, 1, 0, \ldots$ ist die Folge der Partialsummen.
- Zwei wichtige Reihen:
  - $\sum_k \frac{1}{k} = \infty$.

Konvergenz unendlicher Reihen

Wir sagen, dass die unendliche Summe $\sum_{k=0}^{\infty} a_k$ gegen S konvergiert, wenn die Folge (S_n) mit $S_n = \sum_{k=0}^{n} a_k$ gegen S konvergiert.

- Die Reihe $\sum_k (-1)^k$ ist divergent, denn $1, 0, 1, 0 \ldots, 1, 0, \ldots$ ist die Folge der Partialsummen.
- Zwei wichtige Reihen:
  - $\sum_k \frac{1}{k} = \infty$.
  - $\sum_k \frac{1}{k^2}$

Konvergenz unendlicher Reihen

Wir sagen, dass die unendliche Summe $\sum_{k=0}^{\infty} a_k$ gegen S konvergiert, wenn die Folge (S_n) mit $S_n = \sum_{k=0}^{n} a_k$ gegen S konvergiert.

- Die Reihe $\sum_k (-1)^k$ ist divergent, denn $1, 0, 1, 0 \ldots, 1, 0, \ldots$ ist die Folge der Partialsummen.
- Zwei wichtige Reihen:
  - $\sum_k \frac{1}{k} = \infty$.
  - $\sum_k \frac{1}{k^2}$ konvergiert gegen eine reelle Zahl.

Konvergenz unendlicher Reihen

Wir sagen, dass die unendliche Summe $\sum_{k=0}^{\infty} a_k$ gegen S konvergiert, wenn die Folge (S_n) mit $S_n = \sum_{k=0}^{n} a_k$ gegen S konvergiert.

- Die Reihe $\sum_k (-1)^k$ ist divergent, denn $1, 0, 1, 0 \ldots, 1, 0, \ldots$ ist die Folge der Partialsummen.
- Zwei wichtige Reihen:
  - $\sum_k \frac{1}{k} = \infty$.
  - $\sum_k \frac{1}{k^2}$ konvergiert gegen eine reelle Zahl.
- Was können wir über die „Mutter aller Reihen“, nämlich die geometrische Reihe

$$\sum_k x^k$$

sagen?

Die geometrische Reihe

Wir wissen: $\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$.

Die geometrische Reihe

Wir wissen: $\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$.

- $\lim_n \frac{x^{n+1}-1}{x-1} = \frac{1}{x-1}$, falls $|x| < 1$.

Die geometrische Reihe

Wir wissen: $\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$.

- $\lim_n \frac{x^{n+1}-1}{x-1} = \frac{1}{x-1}$, falls $|x| < 1$.
- $\lim_n \frac{x^{n+1}-1}{x-1} = \infty$, falls $x \geq 1$

Die geometrische Reihe

Wir wissen: $\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$.

- $\lim_n \frac{x^{n+1}-1}{x-1} = \frac{1}{x-1}$, falls $|x| < 1$.
- $\lim_n \frac{x^{n+1}-1}{x-1} = \infty$, falls $x \geq 1$ und
- $\sum_k x^k$ divergiert für $x \leq -1$.

Die geometrische Reihe

Wir wissen: $\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$.

- $\lim_n \frac{x^{n+1}-1}{x-1} = \frac{1}{x-1}$, falls $|x| < 1$.
- $\lim_n \frac{x^{n+1}-1}{x-1} = \infty$, falls $x \geq 1$ und
- $\sum_k x^k$ divergiert für $x \leq -1$.

Falls $|x| < 1$ gilt $\sum_k x^k = \frac{1}{x-1}$.

Achilles und die Schildkröte

- Achilles veranstaltet einen Wettlauf mit einer (ziemlich schnellen) Schildkröte.
  - Achilles läuft zehnmal schneller als die Schildkröte, aber
  - die Schildkröte erhält einen Vorsprung von 10 Ellen.

Achilles und die Schildkröte

- Achilles veranstaltet einen Wettlauf mit einer (ziemlich schnellen) Schildkröte.
  - Achilles läuft zehnmal schneller als die Schildkröte, aber
  - die Schildkröte erhält einen Vorsprung von 10 Ellen.
- Die Schildkröte und Achilles starten zur gleichen Zeit.
  - Hat Achilles die ersten 10 Ellen durcheilt, so ist die Schildkröte um eine Elle vorangekommen.
  - Hat Achilles diese Elle zurückgelegt, beträgt der Vorsprung der Schildkröte immer noch $1/10$ Ellen.
  - Der Vorsprung der Schildkröte wird zwar immer kleiner, aber er wird nie Null!?

Achilles und die Schildkröte

- Achilles veranstaltet einen Wettlauf mit einer (ziemlich schnellen) Schildkröte.
  - Achilles läuft zehnmal schneller als die Schildkröte, aber
  - die Schildkröte erhält einen Vorsprung von 10 Ellen.
- Die Schildkröte und Achilles starten zur gleichen Zeit.
  - Hat Achilles die ersten 10 Ellen durcheilt, so ist die Schildkröte um eine Elle vorangekommen.
  - Hat Achilles diese Elle zurückgelegt, beträgt der Vorsprung der Schildkröte immer noch $1/10$ Ellen.
  - Der Vorsprung der Schildkröte wird zwar immer kleiner, aber er wird nie Null!?

An welcher Stelle, falls überhaupt, holt Achilles die Schildkröte ein?

Positive Reihen

Eine positive Reihe konvergiert genau dann, wenn die Folge ihrer Partialsummen beschränkt ist.

Positive Reihen

Eine positive Reihe konvergiert genau dann, wenn die Folge ihrer Partialsummen beschränkt ist.

- Wir wissen: Eine monoton wachsende Folge konvergiert genau dann, wenn sie beschränkt ist.

Positive Reihen

Eine positive Reihe konvergiert genau dann, wenn die Folge ihrer Partialsummen beschränkt ist.

- Wir wissen: Eine monoton wachsende Folge konvergiert genau dann, wenn sie beschränkt ist.
- Eine positive Reihe definiert eine monoton wachsende Folge:

  Die Reihe konvergiert also genau dann, wenn ihre Partialsummen beschränkt sind.

Das Quotientenkriterium

Die Reihe $\sum a_k$ konvergiert genau dann, wenn es eine reelle Zahl $\theta < 1$ gibt, so dass das Quotientenkriterium (Cauchy 1821)

$$|\frac{a_k}{a_{k-1}}| \leq \theta$$

erfüllt ist.

Das Quotientenkriterium

Die Reihe $\sum a_k$ konvergiert genau dann, wenn es eine reelle Zahl $\theta < 1$ gibt, so dass das Quotientenkriterium (Cauchy 1821)

$$|\frac{a_k}{a_{k-1}}| \leq \theta$$

erfüllt ist.

- Wir zeigen, dass $|a_k| \leq |a_0| \cdot \theta^k$ gilt.

Das Quotientenkriterium

Die Reihe $\sum a_k$ konvergiert genau dann, wenn es eine reelle Zahl $\theta < 1$ gibt, so dass das Quotientenkriterium (Cauchy 1821)

$$|\frac{a_k}{a_{k-1}}| \leq \theta$$

erfüllt ist.

- Wir zeigen, dass $|a_k| \leq |a_0| \cdot \theta^k$ gilt.
  - Alle Partialsummen $\sum_{k=0}^{n} |a_k|$ sind durch $\sum_{k=0}^{n} |a_0| \cdot \theta^k$ beschränkt.
  - Konvergenz folgt aus der Konvergenz für die geometrische Reihe.

Das Quotientenkriterium

Die Reihe $\sum a_k$ konvergiert genau dann, wenn es eine reelle Zahl $\theta < 1$ gibt, so dass das Quotientenkriterium (Cauchy 1821)

$$\left|\frac{a_k}{a_{k-1}}\right| \leq \theta$$

erfüllt ist.

- Wir zeigen, dass $|a_k| \leq |a_0| \cdot \theta^k$ gilt.
  - Alle Partialsummen $\sum_{k=0}^{n} |a_k|$ sind durch $\sum_{k=0}^{n} |a_0| \cdot \theta^k$ beschränkt.
  - Konvergenz folgt aus der Konvergenz für die geometrische Reihe.
- $|a_k| \leq \theta |a_{k-1}|$

Das Quotientenkriterium

Die Reihe $\sum a_k$ konvergiert genau dann, wenn es eine reelle Zahl $\theta < 1$ gibt, so dass das Quotientenkriterium (Cauchy 1821)

$$|\frac{a_k}{a_{k-1}}| \leq \theta$$

erfüllt ist.

- Wir zeigen, dass $|a_k| \leq |a_0| \cdot \theta^k$ gilt.
  - Alle Partialsummen $\sum_{k=0}^{n} |a_k|$ sind durch $\sum_{k=0}^{n} |a_0| \cdot \theta^k$ beschränkt.
  - Konvergenz folgt aus der Konvergenz für die geometrische Reihe.
- $|a_k| \leq \theta |a_{k-1}| \leq \theta^2 |a_{k-2}|$

Das Quotientenkriterium

Die Reihe $\sum a_k$ konvergiert genau dann, wenn es eine reelle Zahl $\theta < 1$ gibt, so dass das Quotientenkriterium (Cauchy 1821)

$$|\frac{a_k}{a_{k-1}}| \leq \theta$$

erfüllt ist.

- Wir zeigen, dass $|a_k| \leq |a_0| \cdot \theta^k$ gilt.
  - Alle Partialsummen $\sum_{k=0}^{n} |a_k|$ sind durch $\sum_{k=0}^{n} |a_0| \cdot \theta^k$ beschränkt.
  - Konvergenz folgt aus der Konvergenz für die geometrische Reihe.
- $|a_k| \leq \theta |a_{k-1}| \leq \theta^2 |a_{k-2}| \leq \ldots \leq \theta^k a_0$ und fertig.

Das Quotientenkriterium

Die Reihe $\sum a_k$ konvergiert genau dann, wenn es eine reelle Zahl $\theta < 1$ gibt, so dass das Quotientenkriterium (Cauchy 1821)

$$|\frac{a_k}{a_{k-1}}| \leq \theta$$

erfüllt ist.

- Wir zeigen, dass $|a_k| \leq |a_0| \cdot \theta^k$ gilt.
  - Alle Partialsummen $\sum_{k=0}^{n} |a_k|$ sind durch $\sum_{k=0}^{n} |a_0| \cdot \theta^k$ beschränkt.
  - Konvergenz folgt aus der Konvergenz für die geometrische Reihe.
- $|a_k| \leq \theta |a_{k-1}| \leq \theta^2 |a_{k-2}| \leq \ldots \leq \theta^k a_0$ und fertig.

Warum konvergiert $\sum_k \frac{x^k}{k!}$ für jede reelle Zahl x?

Das Integralkriterium

Die Funktion $f : \mathbb{R} \to \mathbb{R}$ sei gegeben.

Die Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ sei gegeben.
(a) Wenn f monoton wachsend ist, dann gilt

$$\int_0^n f(x)dx \leq$$

Das Integralkriterium

Die Funktion $f : \mathbb{R} \to \mathbb{R}$ sei gegeben.
(a) Wenn f monoton wachsend ist, dann gilt

$$\int_0^n f(x)dx \leq \sum_{i=1}^{n} f(i) \leq$$

Das Integralkriterium

Die Funktion $f : \mathbb{R} \to \mathbb{R}$ sei gegeben.
(a) Wenn f monoton wachsend ist, dann gilt

$$\int_0^n f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_1^{n+1} f(x)dx.$$

Das Integralkriterium

Die Funktion $f : \mathbb{R} \to \mathbb{R}$ sei gegeben.
(a) Wenn f monoton wachsend ist, dann gilt

$$\int_0^n f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_1^{n+1} f(x)dx.$$

(b) Wenn f monoton fallend ist, dann gilt

$$\int_1^{n+1} f(x)dx \leq$$

Das Integralkriterium

Die Funktion $f : \mathbb{R} \to \mathbb{R}$ sei gegeben.

(a) Wenn f monoton wachsend ist, dann gilt

$$\int_0^n f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_1^{n+1} f(x)dx.$$

(b) Wenn f monoton fallend ist, dann gilt

$$\int_1^{n+1} f(x)dx \leq \sum_{i=1}^{n} f(i) \leq$$

Das Integralkriterium

Die Funktion $f : \mathbb{R} \to \mathbb{R}$ sei gegeben.
(a) Wenn f monoton wachsend ist, dann gilt

$$\int_0^n f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_1^{n+1} f(x)dx.$$

(b) Wenn f monoton fallend ist, dann gilt

$$\int_1^{n+1} f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_0^{n} f(x)dx.$$

Das Integralkriterium

Die Funktion $f : \mathbb{R} \to \mathbb{R}$ sei gegeben.
(a) Wenn f monoton wachsend ist, dann gilt

$$\int_0^n f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_1^{n+1} f(x)dx.$$

(b) Wenn f monoton fallend ist, dann gilt

$$\int_1^{n+1} f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_0^{n} f(x)dx.$$

Zwei wichtige Konsequenzen:

Das Integralkriterium

Die Funktion $f : \mathbb{R} \to \mathbb{R}$ sei gegeben.
(a) Wenn f monoton wachsend ist, dann gilt

$$\int_0^n f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_1^{n+1} f(x)dx.$$

(b) Wenn f monoton fallend ist, dann gilt

$$\int_1^{n+1} f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_0^{n} f(x)dx.$$

Zwei wichtige Konsequenzen:

- $\sum_{i=1}^{n} \frac{1}{i} =$

Das Integralkriterium

Die Funktion $f : \mathbb{R} \to \mathbb{R}$ sei gegeben.
(a) Wenn f monoton wachsend ist, dann gilt

$$\int_0^n f(x)dx \leq \sum_{i=1}^n f(i) \leq \int_1^{n+1} f(x)dx.$$

(b) Wenn f monoton fallend ist, dann gilt

$$\int_1^{n+1} f(x)dx \leq \sum_{i=1}^n f(i) \leq \int_0^n f(x)dx.$$

Zwei wichtige Konsequenzen:

- $\sum_{i=1}^{n} \frac{1}{i} = \Theta(\ln(n))$.

Das Integralkriterium

Die Funktion $f : \mathbb{R} \to \mathbb{R}$ sei gegeben.
(a) Wenn f monoton wachsend ist, dann gilt

$$\int_0^n f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_1^{n+1} f(x)dx.$$

(b) Wenn f monoton fallend ist, dann gilt

$$\int_1^{n+1} f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_0^n f(x)dx.$$

Zwei wichtige Konsequenzen:

- $\sum_{i=1}^{n} \frac{1}{i} = \Theta(\ln(n))$.
- $\sum_{i=1}^{n} i^k =$

Die Funktion $f : \mathbb{R} \to \mathbb{R}$ sei gegeben.
(a) Wenn f monoton wachsend ist, dann gilt

$$\int_0^n f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_1^{n+1} f(x)dx.$$

(b) Wenn f monoton fallend ist, dann gilt

$$\int_1^{n+1} f(x)dx \leq \sum_{i=1}^{n} f(i) \leq \int_0^{n} f(x)dx.$$

Zwei wichtige Konsequenzen:

- $\sum_{i=1}^{n} \frac{1}{i} = \Theta(\ln(n))$.
- $\sum_{i=1}^{n} i^k = \Theta(n^{k+1})$, falls $k \geq 0$.

Wie berechnet man a^x?

- Wenn der **Exponent** $x = m$ eine natürliche Zahl ist, dann ist die **Basis** a genau x Mal zu multplizieren.

Wie berechnet man a^x?

- Wenn der **Exponent** $x = m$ eine natürliche Zahl ist, dann ist die **Basis** a genau x Mal zu multplizieren.
- Wenn $x = 1/n$, dann ist die $n$te Wurzel von a zu bestimmen.

Wie berechnet man a^x?

- Wenn der **Exponent** $x = m$ eine natürliche Zahl ist, dann ist die **Basis** a genau x Mal zu multplizieren.
- Wenn $x = 1/n$, dann ist die $n$te Wurzel von a zu bestimmen.
- Und wenn x eine reelle Zahl ist?
  - Wir fordern $f(x + y) = a^{x+y}$

Wie berechnet man a^x?

- Wenn der **Exponent** $x = m$ eine natürliche Zahl ist, dann ist die **Basis** a genau x Mal zu multplizieren.
- Wenn $x = 1/n$, dann ist die $n$te Wurzel von a zu bestimmen.
- Und wenn x eine reelle Zahl ist?
  - Wir fordern $f(x+y) = a^{x+y} \stackrel{!}{=} a^x a^y$

Wie berechnet man a^x?

- Wenn der **Exponent** $x = m$ eine natürliche Zahl ist, dann ist die **Basis** a genau x Mal zu multplizieren.
- Wenn $x = 1/n$, dann ist die $n$te Wurzel von a zu bestimmen.
- Und wenn x eine reelle Zahl ist?
  - Wir fordern $f(x+y) = a^{x+y} \stackrel{!}{=} a^x a^y = f(x) \cdot f(y)$.

Wie berechnet man a^x?

- Wenn der **Exponent** $x = m$ eine natürliche Zahl ist, dann ist die **Basis** a genau x Mal zu multplizieren.
- Wenn $x = 1/n$, dann ist die $n$te Wurzel von a zu bestimmen.
- Und wenn x eine reelle Zahl ist?
  - Wir fordern $f(x+y) = a^{x+y} \stackrel{!}{=} a^x a^y = f(x) \cdot f(y)$.
  - Aus dieser Forderung folgt wie gewünscht

$$f(m/n) = f(1/n)^m$$

Wie berechnet man a^x?

- Wenn der **Exponent** $x = m$ eine natürliche Zahl ist, dann ist die **Basis** a genau x Mal zu multplizieren.
- Wenn $x = 1/n$, dann ist die $n$te Wurzel von a zu bestimmen.
- Und wenn x eine reelle Zahl ist?
  - Wir fordern $f(x+y) = a^{x+y} \stackrel{!}{=} a^x a^y = f(x) \cdot f(y)$.
  - Aus dieser Forderung folgt wie gewünscht

$$f(m/n) = f(1/n)^m = (f(1)^{1/n})^m$$

Wie berechnet man a^x?

- Wenn der **Exponent** $x = m$ eine natürliche Zahl ist, dann ist die **Basis** a genau x Mal zu multplizieren.
- Wenn $x = 1/n$, dann ist die $n$te Wurzel von a zu bestimmen.
- Und wenn x eine reelle Zahl ist?
  - Wir fordern $f(x+y) = a^{x+y} \stackrel{!}{=} a^x a^y = f(x) \cdot f(y)$.
  - Aus dieser Forderung folgt wie gewünscht

$$f(m/n) = f(1/n)^m = (f(1)^{1/n})^m = f(1)^{m/n}$$

Wie berechnet man a^x?

- Wenn der **Exponent** $x = m$ eine natürliche Zahl ist, dann ist die **Basis** a genau x Mal zu multplizieren.
- Wenn $x = 1/n$, dann ist die $n$te Wurzel von a zu bestimmen.
- Und wenn x eine reelle Zahl ist?
  - Wir fordern $f(x + y) = a^{x+y} \stackrel{!}{=} a^x a^y = f(x) \cdot f(y)$.
  - Aus dieser Forderung folgt wie gewünscht

$$f(m/n) = f(1/n)^m = (f(1)^{1/n})^m = f(1)^{m/n} = a^{m/n}.$$

Wie berechnet man a^x?

- Wenn der **Exponent** $x = m$ eine natürliche Zahl ist, dann ist die **Basis** a genau x Mal zu multplizieren.
- Wenn $x = 1/n$, dann ist die $n$te Wurzel von a zu bestimmen.
- Und wenn x eine reelle Zahl ist?
  - Wir fordern $f(x+y) = a^{x+y} \stackrel{!}{=} a^x a^y = f(x) \cdot f(y)$.
  - Aus dieser Forderung folgt wie gewünscht

$$f(m/n) = f(1/n)^m = (f(1)^{1/n})^m = f(1)^{m/n} = a^{m/n}.$$

  - Wenn f stetig ist: $f(x)$ ist der Grenzwert der Folge a^{x_n} für jede gegen x konvergente Folge (x_n) rationaler Zahlen.

Wie berechnet man a^x?

- Wenn der **Exponent** $x = m$ eine natürliche Zahl ist, dann ist die **Basis** a genau x Mal zu multplizieren.
- Wenn $x = 1/n$, dann ist die $n$te Wurzel von a zu bestimmen.
- Und wenn x eine reelle Zahl ist?
  - Wir fordern $f(x+y) = a^{x+y} \stackrel{!}{=} a^x a^y = f(x) \cdot f(y)$.
  - Aus dieser Forderung folgt wie gewünscht

$$f(m/n) = f(1/n)^m = (f(1)^{1/n})^m = f(1)^{m/n} = a^{m/n}.$$

  - Wenn f stetig ist: $f(x)$ ist der Grenzwert der Folge a^{x_n} für jede gegen x konvergente Folge (x_n) rationaler Zahlen.

Wie sieht die Funktion f aus?

Die Exponentialfunktion

Setze

$$f(x) = e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

Dann gilt:

(a) Die Summe konvergiert für jede reelle Zahl x,

(b) und für alle reellen Zahlen a und b ist
$e^{a+b} = e^a \cdot e^b$ wie auch $(e^a)^b = e^{ab}$.

Die Exponentialfunktion

Setze

$$f(x) = e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

Dann gilt:

(a) Die Summe konvergiert für jede reelle Zahl x,

(b) und für alle reellen Zahlen a und b ist
$e^{a+b} = e^a \cdot e^b$ wie auch $(e^a)^b = e^{ab}$.

- Warum? Konvergenz zeigen wir später. Für (b) siehe Skript.

Die Exponentialfunktion

Setze

$$f(x) = e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

Dann gilt:

(a) Die Summe konvergiert für jede reelle Zahl x,

(b) und für alle reellen Zahlen a und b ist $e^{a+b} = e^a \cdot e^b$ wie auch $(e^a)^b = e^{ab}$.

- Warum? Konvergenz zeigen wir später. Für (b) siehe Skript.
- Wie löst man das umgekehrte Problem: Wir kennen y und möchten x bestimmen, so dass $y = a^x$ gilt?

Der Logarithmus

Seien $a > 1$ und $x > 0$ reelle Zahlen.

- Der natürliche Logarithmus $\ln(x)$ von x stimmt mit y genau dann überein, wenn $e^y = x$ gilt.
- Der Logarithmus $\log_a(x)$ zur Basis a stimmt mit z genau dann überein, wenn $a^z = x$.

Der Logarithmus

Seien $a > 1$ und $x > 0$ reelle Zahlen.

- Der natürliche Logarithmus $\ln(x)$ von x stimmt mit y genau dann überein, wenn $e^y = x$ gilt.
- Der Logarithmus $\log_a(x)$ zur Basis a stimmt mit z genau dann überein, wenn $a^z = x$.

- Wir müssen a^x definieren:

$$a^x = e^{\ln(a)\cdot x}.$$

Der Logarithmus

Seien $a > 1$ und $x > 0$ reelle Zahlen.

- Der natürliche Logarithmus $\ln(x)$ von x stimmt mit y genau dann überein, wenn $e^y = x$ gilt.
- Der Logarithmus $\log_a(x)$ zur Basis a stimmt mit z genau dann überein, wenn $a^z = x$.

- Wir müssen a^x definieren:

$$a^x = e^{\ln(a) \cdot x}.$$

- Wieviele Bits hat die Binärdarstellung der natürlichen Zahl n?

Der Logarithmus

Seien $a > 1$ und $x > 0$ reelle Zahlen.

- Der natürliche Logarithmus $\ln(x)$ von x stimmt mit y genau dann überein, wenn $e^y = x$ gilt.
- Der Logarithmus $\log_a(x)$ zur Basis a stimmt mit z genau dann überein, wenn $a^z = x$.

- Wir müssen a^x definieren:

$$a^x = e^{\ln(a) \cdot x}.$$

- Wieviele Bits hat die Binärdarstellung der natürlichen Zahl n?
- Wie rechnet man mit Logarithmen?

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) =$

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.

(b) $\log_a(x^y) =$

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.

(b) $\log_a(x^y) = y \cdot \log_a(x)$.

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.

(b) $\log_a(x^y) = y \cdot \log_a(x)$.

(c) $a^{\log_a x} =$

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.

(b) $\log_a(x^y) = y \cdot \log_a(x)$.

(c) $a^{\log_a x} = x$.

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.

(b) $\log_a(x^y) = y \cdot \log_a(x)$.

(c) $a^{\log_a x} = x$.

(d) Von Basis *a* zur Basis *b*: $\log_a x =$

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.

(b) $\log_a(x^y) = y \cdot \log_a(x)$.

(c) $a^{\log_a x} = x$.

(d) Von Basis *a* zur Basis *b*: $\log_a x = (\log_a b) \cdot (\log_b x)$.

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.
(b) $\log_a(x^y) = y \cdot \log_a(x)$.
(c) $a^{\log_a x} = x$.
(d) Von Basis *a* zur Basis *b*: $\log_a x = (\log_a b) \cdot (\log_b x)$.

- Was ist $\log_a n^k$?

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.
(b) $\log_a(x^y) = y \cdot \log_a(x)$.
(c) $a^{\log_a x} = x$.
(d) Von Basis a zur Basis b: $\log_a x = (\log_a b) \cdot (\log_b x)$.

- Was ist $\log_a n^k$? $\log_a n^k = k \cdot \log_a n$.

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.
(b) $\log_a(x^y) = y \cdot \log_a(x)$.
(c) $a^{\log_a x} = x$.
(d) Von Basis a zur Basis b: $\log_a x = (\log_a b) \cdot (\log_b x)$.

- Was ist $\log_a n^k$? $\log_a n^k = k \cdot \log_a n$.
- $4^{\log_2 n} =$

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.
(b) $\log_a(x^y) = y \cdot \log_a(x)$.
(c) $a^{\log_a x} = x$.
(d) Von Basis *a* zur Basis *b*: $\log_a x = (\log_a b) \cdot (\log_b x)$.

- Was ist $\log_a n^k$? $\log_a n^k = k \cdot \log_a n$.
- $4^{\log_2 n} = (2^2)^{\log_2 n} =$

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.

(b) $\log_a(x^y) = y \cdot \log_a(x)$.

(c) $a^{\log_a x} = x$.

(d) Von Basis *a* zur Basis *b*: $\log_a x = (\log_a b) \cdot (\log_b x)$.

- Was ist $\log_a n^k$? $\log_a n^k = k \cdot \log_a n$.
- $4^{\log_2 n} = (2^2)^{\log_2 n} = 2^{2 \log_2 n} =$

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.
(b) $\log_a(x^y) = y \cdot \log_a(x)$.
(c) $a^{\log_a x} = x$.
(d) Von Basis *a* zur Basis *b*: $\log_a x = (\log_a b) \cdot (\log_b x)$.

- Was ist $\log_a n^k$? $\log_a n^k = k \cdot \log_a n$.
- $4^{\log_2 n} = (2^2)^{\log_2 n} = 2^{2\log_2 n} = (2^{\log_2 n})^2 = n^2$.

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.
(b) $\log_a(x^y) = y \cdot \log_a(x)$.
(c) $a^{\log_a x} = x$.
(d) Von Basis a zur Basis b: $\log_a x = (\log_a b) \cdot (\log_b x)$.

- Was ist $\log_a n^k$? $\log_a n^k = k \cdot \log_a n$.
- $4^{\log_2 n} = (2^2)^{\log_2 n} = 2^{2\log_2 n} = (2^{\log_2 n})^2 = n^2$.

Was ist $b^{\log_a x}$?

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.
(b) $\log_a(x^y) = y \cdot \log_a(x)$.
(c) $a^{\log_a x} = x$.
(d) Von Basis a zur Basis b: $\log_a x = (\log_a b) \cdot (\log_b x)$.

- Was ist $\log_a n^k$? $\log_a n^k = k \cdot \log_a n$.
- $4^{\log_2 n} = (2^2)^{\log_2 n} = 2^{2 \log_2 n} = (2^{\log_2 n})^2 = n^2$.

Was ist $b^{\log_a x}$?

- $\log_a x = (\log_a b) \cdot (\log_b x)$ mit (d).

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.

(b) $\log_a(x^y) = y \cdot \log_a(x)$.

(c) $a^{\log_a x} = x$.

(d) Von Basis *a* zur Basis *b*: $\log_a x = (\log_a b) \cdot (\log_b x)$.

- Was ist $\log_a n^k$? $\log_a n^k = k \cdot \log_a n$.
- $4^{\log_2 n} = (2^2)^{\log_2 n} = 2^{2\log_2 n} = (2^{\log_2 n})^2 = n^2$.

Was ist $b^{\log_a x}$?

- $\log_a x = (\log_a b) \cdot (\log_b x)$ mit (d).
- $b^{\log_a x} = b^{(\log_a b)\cdot(\log_b x)}$

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.

(b) $\log_a(x^y) = y \cdot \log_a(x)$.

(c) $a^{\log_a x} = x$.

(d) Von Basis *a* zur Basis *b*: $\log_a x = (\log_a b) \cdot (\log_b x)$.

- Was ist $\log_a n^k$? $\log_a n^k = k \cdot \log_a n$.
- $4^{\log_2 n} = (2^2)^{\log_2 n} = 2^{2 \log_2 n} = (2^{\log_2 n})^2 = n^2$.

Was ist $b^{\log_a x}$?

- $\log_a x = (\log_a b) \cdot (\log_b x)$ mit (d).
- $b^{\log_a x} = b^{(\log_a b) \cdot (\log_b x)} = (b^{\log_b x})^{\log_a b}$

Rechnen mit Logarithmen

(a) $\log_a(x \cdot y) = \log_a x + \log_a y$.
(b) $\log_a(x^y) = y \cdot \log_a(x)$.
(c) $a^{\log_a x} = x$.
(d) Von Basis a zur Basis b: $\log_a x = (\log_a b) \cdot (\log_b x)$.

- Was ist $\log_a n^k$? $\log_a n^k = k \cdot \log_a n$.
- $4^{\log_2 n} = (2^2)^{\log_2 n} = 2^{2\log_2 n} = (2^{\log_2 n})^2 = n^2$.

Was ist $b^{\log_a x}$?

- $\log_a x = (\log_a b) \cdot (\log_b x)$ mit (d).
- $b^{\log_a x} = b^{(\log_a b)\cdot(\log_b x)} = (b^{\log_b x})^{\log_a b} = x^{\log_a b}$.