Informatik III

Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Christian Schindelhauer
Wintersemester 2006/07
18. Vorlesung
22.12.2006

Komplexitätstheorie - Zeitklassen

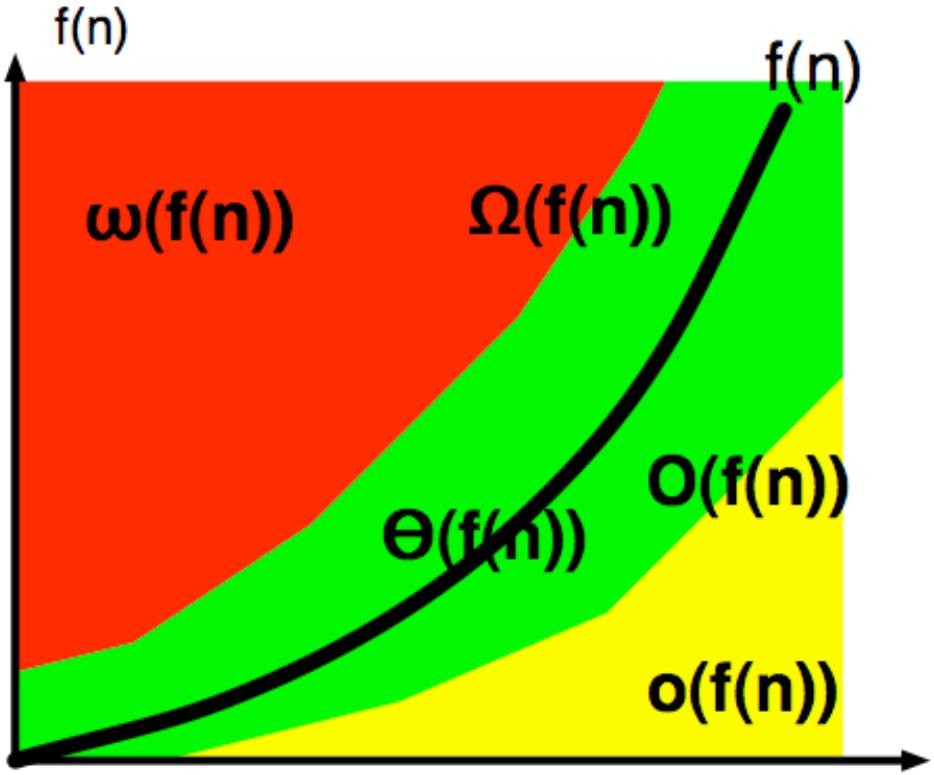
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Komplexitätsmaße**
  - Wiederholung: O,o,ω,Θ,Ω
  - Laufzeitanalyse
- **Die Komplexitätsklassen TIME,**
  - DTIME, NTIME
  - P mit Beispielen
  - NP mit Beispielen
- **Das Cook-Levin-Theorem**
  - Polynomial-Zeit-Reduktion
  - Reduktionen zwischen 3SAT und Clique
  - NP-vollständigkeit
  - SAT ist NP-vollständig
- **Weitere NP-vollständige Probleme**
  - Knotenüberdeckung (Vertex-Cover)
  - Das Hamiltonsche Pfadproblem
  - Das ungerichtete Hamiltonsche Pfadproblem
  - Das Teilsummenproblem

Die asymptotischen Wachstumsklassen

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

$$
\begin{aligned}
O(g) &:= \{f \mid \exists k \in \mathbb{R} \quad f \leq_{ae} k \cdot g\}, \\
o(g) &:= \{f \mid \forall k \in \mathbb{R}^+ \quad k \cdot f \leq_{ae} g\}, \\
\omega(g) &:= \{f \mid \forall k \in \mathbb{R}^+ \quad k \cdot g \leq_{ae} f\}, \\
\Omega(g) &:= \{f \mid \exists k \in \mathbb{R} \quad g \leq_{ae} k \cdot f\}, \\
\Theta(g) &:= O(g) \quad \cap \quad \Omega(g).
\end{aligned}
$$



Korrekte und “schlampige” Notationen

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Elemente der Wachstumsklassen sind Funktionen:**

$$n \mapsto f(n)$$

- **Diese werden so abgekürzt:**

$$\mathcal{N} := n \mapsto n$$

$$\mathcal{N}^2 := n \mapsto n^2$$

$$2^{\mathcal{N}} := n \mapsto 2^n$$

- **Eine Funktion ist Element einer Wachstumsklasse**

$$\mathcal{N}^2 \in o(2^{\mathcal{N}})$$

$$(n \mapsto n^2) \in o(n \mapsto 2^n)$$

- **Meistens schreibt man einfach nur f(n) und nimmt implizit an, dass n das Argument ist**

$$f(n)$$

- **Man schreibt einfach nur**

$$n$$

$$n^2$$

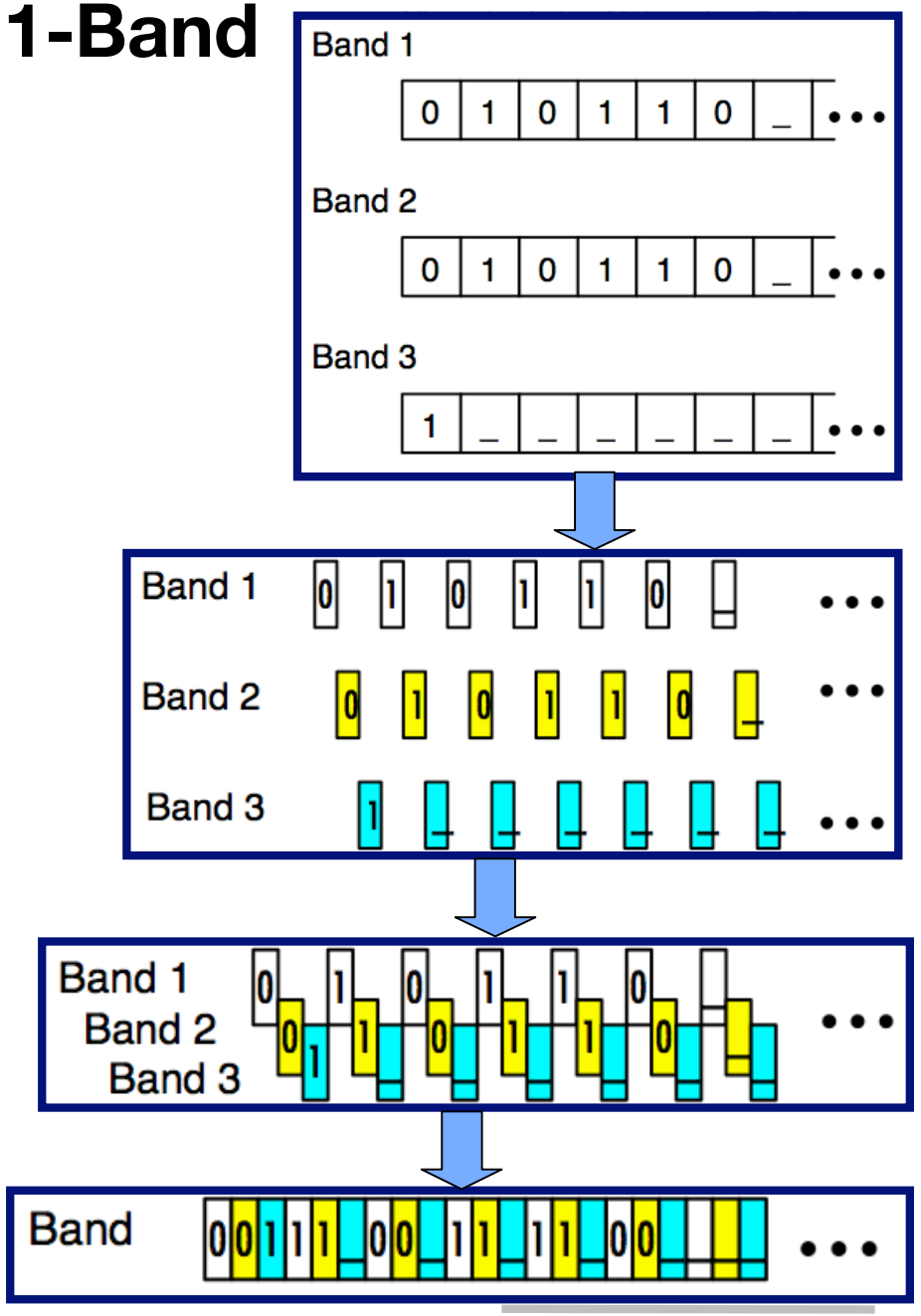
$$2^n$$

- **Statt dem Element-Zeichen wird “=“ geschrieben:**

$$n^2 = o(2^n)$$

k-Band-DTMs → 1-Band DTMs

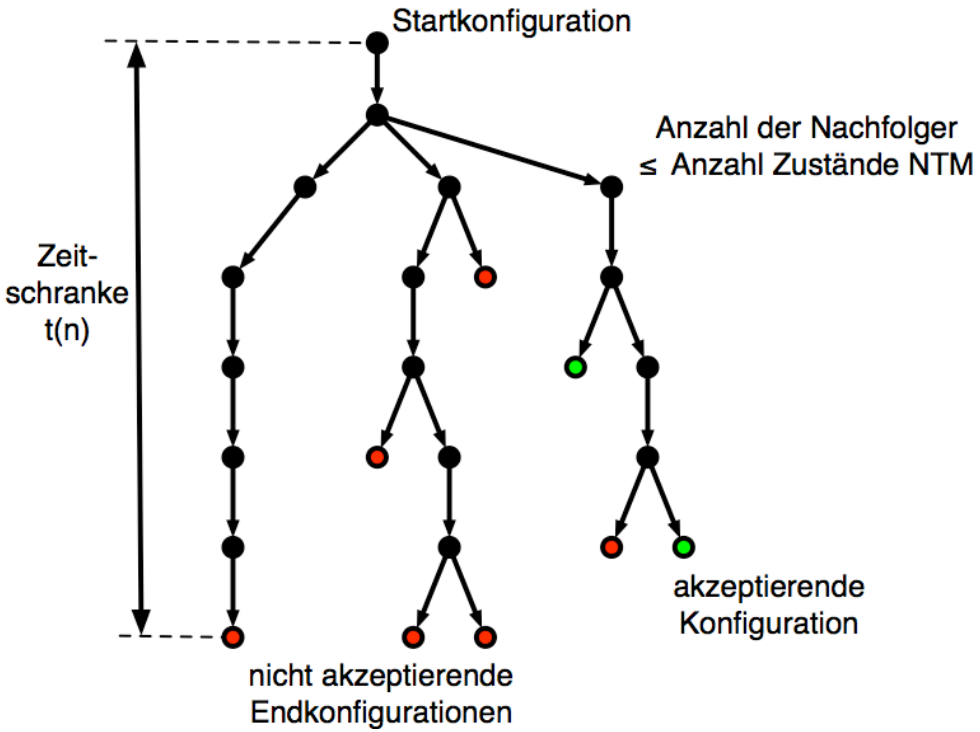
- **Theorem**
  - **$TIME_k(t(n)) \subseteq TIME_1(O(t^2(n))$,** d.h.
  - Für $t(n)=\Omega(n)$ kann jede Berechnung einer k-Band-DTM von einer 1-Band DTM in Zeit $O(t^2(n))$ berechnet werden.
- **Beweis(anfang):**
  - Betrachte k-Band-DTM M mit Arbeitsalphabet Γ
  - und konstruiere 1-Band-DTM mit Alphabet $\Gamma \times \{\_, kopf\}$,
    - Speichere das i.-te Symbol von Band j an der Position j + i k.
    - Verwende Markierung *kopf* nur wenn der Kopf der k-Band-TM an der entsprechenden Stelle steht.
  - ...



Nichtdeterministische Zeitkomplexitätsklassen

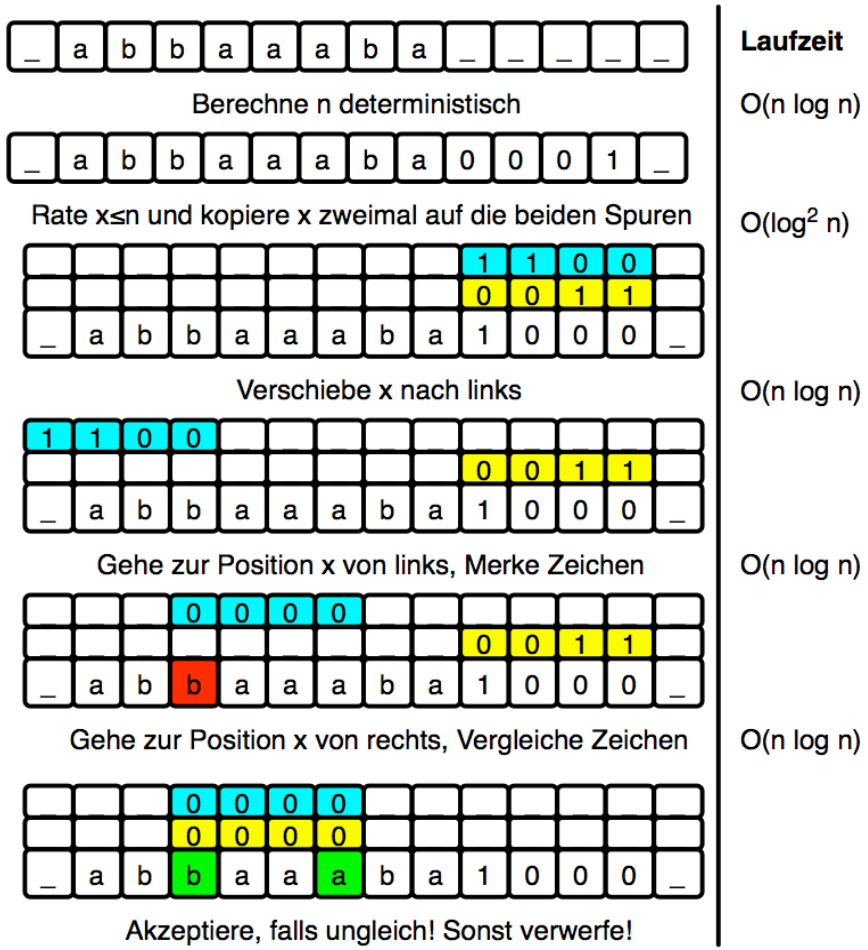
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Definition**
  - Eine NTM ist t-Zeit-beschränkt, wenn für eine Eingabe der Länge n jede nichtdeterministische Berechnung höchstens t(n) Schritte benötigt.
- **Definition**
  - Sei $t:N \rightarrow R^+$ eine Funktion.
  - Die Zeitkomplexitätsklasse **NTIME(t(n))** ist die Menge aller Sprachen,
    - die von einer **nicht**deterministischen O(t(n))-Zeit-Turing-Maschine entschieden werden.
  - Wird die Anzahl der Bänder auf k beschränkt, schreiben wir **$NTIME_{k\text{-}Band}(t(n))$** oder einfach **$NTIME_k(t(n))$.**



Beispiel

- ***Nicht-Palindrom*** **= {w ∈ {a,b}* | w≠w^{rev} }**
- **Es gilt:**
  Nicht-Palindrom **∈ NTIME$_{1\text{-Band}}$(n log n),**
  - da es eine O(n log n) -Zeit 1-Band-NTM die *Nicht-Palindrom* in Laufzeit O(n log n) löst.
- **Lösung:**
  - Rate Position x≤n
    - Dafür berechne zuerst deterministisch erstmal die Eingabegröße
  - Lies Zeichen an Position x und n-x
    - Speichere Positionszähler auf Extra-Spur
    - Verschiebe Positionszähler immer um eins
  - Merke das Zeichen jeweils im endlichen Speicher
  - Falls gilt $w_x \neq w_{n-x}$ akzeptiere, ansonsten verwerfe
- **Beachte: Es gilt**
  Nicht-Palindrom **∉ TIME$_{1\text{-Band}}$(n log n).**



Berechnung einer 1-Band-NTM durch eine 1-Band-DTM

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

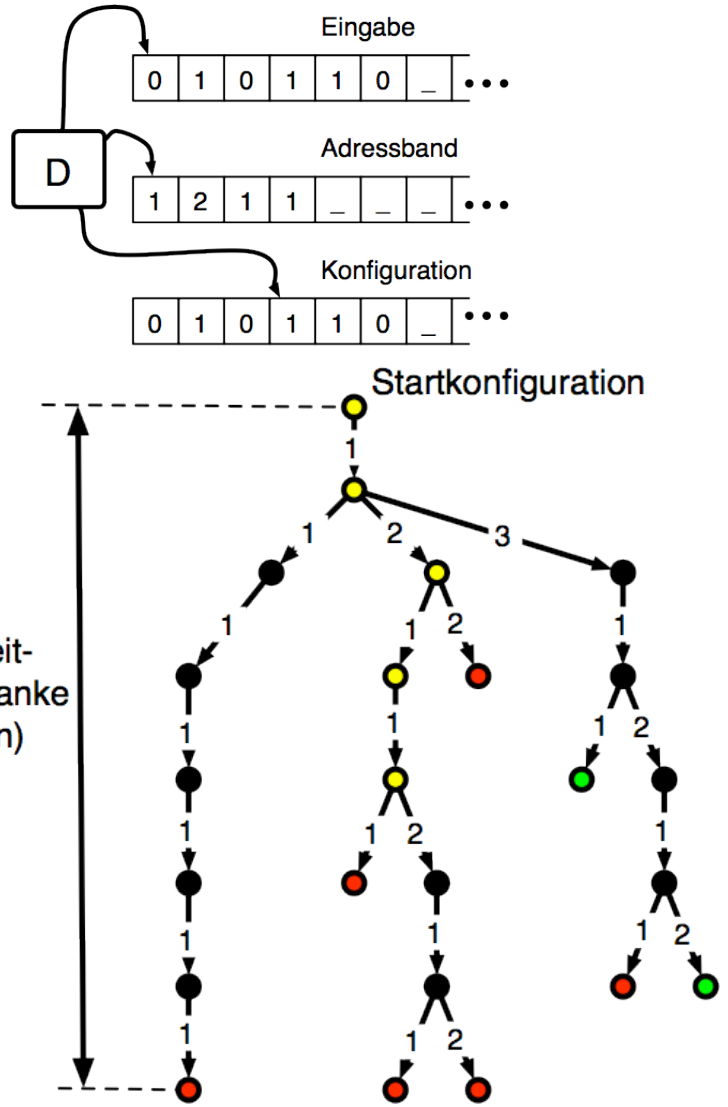
- **Theorem: Für $t(n)=\Omega(n)$**
  - **$NTIME_1(t(n)) \subseteq TIME_1(2^{O(t(n))})$, d.h.**
  - Jede Berechnung einer t-Zeit 1-Band-NTM kann von einer 3-Band DTM in Zeit $2^{O(t(n))}$ durchgeführt werden.
- **Beweis:**
  - Eine Konfiguration einer TM besteht aus
    - der Bandinschrift,
    - der markierten Kopfposition und
    - dem aktuellen Zustand
  - Simuliere 1-Band-NTM mit 3-Band-DTM
    - Speichere auf Band 1 die Eingabe
    - Speichere auf Band 2 die Position der NTM im Berechnungsbaum
    - Verwende Band 3 zum Speichern der aktuellen Konfiguration
  - **Lemma**:
    - Die Anzahl der Knoten im Berechnungsbaum einer t-Zeit NTM mit Zustandsmenge Q ist beschränkt durch $|Q|^{t(n)+1}$.
  - **Fakt**:
    - Jeder Konfigurationsübergang im Berechnungsbaum der NTM kann in einem Schritt auf dem dritten Band berechnet werden.
  - Die 3-Band-DTM wird dann mit einer 1-Band DTM berechnet.

Zeitbeschränkte Simulation einer NTM durch eine DTM

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Beweis:

- Simulator DTM D:
  - Eingabeband
  - Adressband
  - Konfigurationsband
- Zähle per Tiefensuche alle Knoten des Berechnungsbaums im Adressband auf. Sortierung:
  - Lexikographisch
    - 1111111111
    - 1111111112
    - ...
    - 3333333333
- Für jeden Knoten des Berechnungsbaums:
  - Initialisiere Startkonfiguration
  - Führe dabei den (nun deterministischen) Konfigurationsübergang für jeden Pfad in Zeit t(n) durch
  - Falls simulierte NTM in akz. Zustand, dann halte und akzeptiere
- Falls jeder Knoten abgearbeitet, verwerfe



Laufzeit

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Die Anzahl der Blätter eines Baumes mit Ausgrad d und Tiefe t ist beschränkt durch**

$$1 + d + d^2 + d^3 + \ldots + d^t = \frac{d^{t+1} - 1}{d - 1} \leq d^{t+1}$$

  - Nun ist d=|Q| und t=t(n)

- **Die Berechnung bis zur Tiefe t(n) benötigt jeweils t(n) Schritte und n Schritte um die Startkonfiguration wiederherzustellen.**
- **Simulation der 1-Band NTM mit der 3-Band-DTM benötigt höchstens Zeit**

$$|Q|^{t(n)+1} t(n) = 2^{(\log |Q|)(t(n)+1)+\log t(n)} = 2^{O(t(n)}$$

  - da |Q| eine Konstante ist
- **Simulation der 3-Band-DTM mit einer 1-Band-DTM benötigt Zeit:**

$$\left(2^{O(t(n))}\right)^2 = 2^{2 \cdot O(t(n))} = 2^{O(t(n))}$$

DTM versus NTM
k versus k' Bänder

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Theorem: Für $k,k' \geq 1$, $t(n) = \Omega(n)$**
  - **$TIME_k(t(n)) \subseteq TIME_{k'}(t(n)^2)$**
    - Jede Berechnung einer t-Zeit-k-Band-DTM kann von einer $O(t(n)^2)$-Zeit-k'-Band-DTM berechnet werden.
  - **$NTIME_k(t(n)) \subseteq NTIME_{k'}(t(n)^2)$**
    - Jede Berechnung einer t-Zeit-k-Band-NTM kann von einer $O(t(n)^2)$-Zeit-k'-Band-NTM berechnet werden.
  - **$NTIME_k(t(n)) \subseteq TIME_{k'}(2^{O(t(n))})$**
    - Jede Berechnung einer t-Zeit-k-Band-DTM kann von einer $2^{O(t(n))}$-Zeit-k'-Band-DTM berechnet werden.

Zwei wichtige Komplexitätsklassen

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

P & NP

Zwei wichtige Komplexitätsklassen

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Definition:**
  - $\mathrm{P} = \bigcup_k \mathrm{TIME}(n^k)$
  - $\mathrm{NP} = \bigcup_k \mathrm{NTIME}(n^k)$
- **Noch mal:**
  - P: Klasse aller Sprachen, die von einer Polynom-Zeit DTM entschieden werden
  - NP: Klasse aller Sprachen, die von einer Polynom-Zeit NTM entschieden werden können.

Probleme berechenbar in Polynom-Zeit

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Bitte nicht die Laufzeiten lernen!

- **Binärrechnung** Laufzeit in der Anzahl der Stellen der Binärzahl*
  - Addition $O(n)$
  - Vergleich (<) $O(n)$
  - Multiplikation* $O(n^2)$
  - Matrixmultiplikation zweier n × n-Matrizen mit n-stelligen Einträgen* $O(n^5)$
  - Berechnung der Determinante einer n × n-Matrizen mit n-stelligen Einträgen* $O(n^5)$
  - Division mit Rest* $O(n^2)$
  - Größter gemeinsamer Teiler (ggT)* $O(n^3)$
  - Kleinstes gemeinsame Vielfaches (kgV)* $O(n^3)$

*Laufzeit ist nicht unbedingt optimal

Probleme berechenbar in Polynom-Zeit

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Bitte nicht die Laufzeiten lernen!

- **Operationen auf n Zahlen mit je m Stellen**
  - Minimum O(n m)
  - Maximum O(n m)
  - Median O(n m)
  - Mittelwert O(n m)
  - Sortieren O(n m log n)
    - Sortieren mit Bubble-Sort O(n^2 m)
  - Umdrehen O(n m)
  - Shuffle-Operation O(n m)
    - = Kartenmisch-Operation
  - Permutiere gemäß gegebener Permutation O(n m log n)

*Laufzeit ist nicht unbedingt optimal

Probleme berechenbar in Polynom-Zeit

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Bitte nicht die Laufzeiten lernen!

➢ **Graphenprobleme:**	**Laufzeit: n Knoten/m Kanten Graph ist gegeben als**
Adjazenzliste	
– Gibt es einen Weg von a nach b	O((n+m) log n)
– Kürzester Weg von a nach b*	O(n m log n)
– Gibt es einen Kreis im Graph*	O((n+m) m log n)
– Maximaler Grad	O(m log n)
– Durchmesser*	$O(n^3 m \log n)$
– Zusammenhangskomponenten*	$O((n+m)n^2 \log n)$

*Laufzeit ist nicht unbedingt optimal

NP und P

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Definition:**

$$\mathrm{NP} = \bigcup_k \mathrm{NTIME}(n^k)$$

- **Theorem**

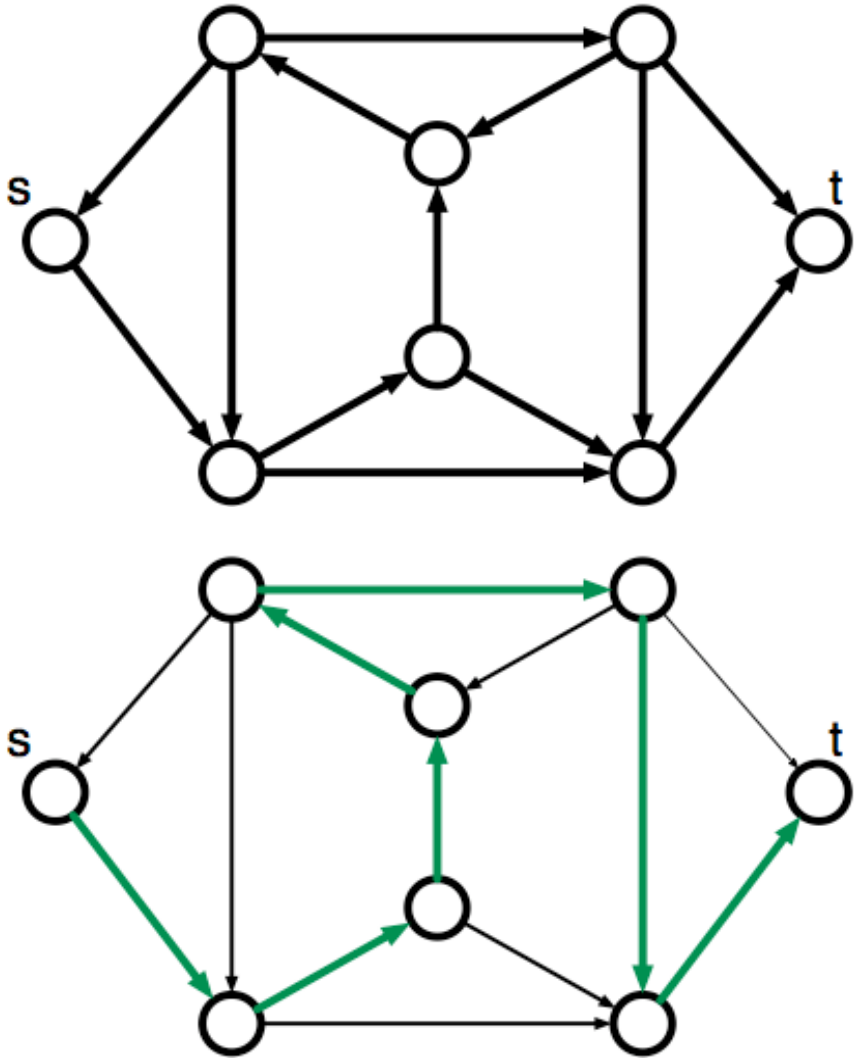
$$\mathrm{P} \subseteq \mathrm{NP}$$

- **Alles was in P ist, ist auch in NP**

Hamiltonsche Pfade

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Definition:** *HAMPATH*
  - Das Hamiltonsche Pfadproblem
    - Geg.:
      - ein gerichteter Graph
      - Zwei Knoten s,t
    - Ges.: existiert ein Hamitonscher Pfad von s nach t
      - d.h. ein gerichteter Pfad, der alle Knoten besucht, aber keine Kante zweimal benutzt
- **Algorithmus für Hamiltonscher Pfad:**
  - Rate eine Permutation $(s,v_1,v_2,...,v_{n-2},t)$
  - Teste, ob Permutation ein Pfad ist
    - falls ja, akzeptiere
    - falls nein, verwerfe
- **Also: HamPath ∈ NP**



Die Nicht-Primzahlen

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Definition:** *COMPOSITES*
  - Geg.: x (als Binärzahl)
  - Ges.: Gibt es ganze Zahlen p,q>1
    - so dass x = p q

COMPOSITES
:= {x | x = p q, für ganze Zahlen p,q>1}

- **NTM für** *COMPOSITES* **:**
  - Rate p,q>1
  - Berechne p q
  - Akzeptiere, falls p q = x
  - Verwerfe sonst

- **Also ist** *COMPOSITES* $\in$ **NP**

Raum für eigene Notizen

Der Verifizierer

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Definition**
  - Ein **Verifizierer** für eine Sprache A ist ein Algorithmus V, wobei
    - A = {w | V akzeptiert <w,c> für ein Wort c}
  - Ein **Polynom-Zeit-Verifizierer** hat eine Laufzeit die durch ein Polynom $O(|w|^k)$ beschränkt ist.
  - Eine Sprache ist in **Polynom-Zeit verifizierbar**, falls sie einen Polynom-Zeit-Verifizierer hat.

- **Theorem**
  - NP beschreibt genau die die Sprachen, die in Polynom-Zeit verifiziert werden können.

- **Beweis:**
  - Nächstes Jahr

Ende der
18. Vorlesung

Frohe Weihnachten!
Mehr Glück im nächsten Jahr!

CoNe Freiburg

Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Christian Schindelhauer
Wintersemester 2006/07
18. Vorlesung
22.12.2006