

## Algorithmen I Tutorium 19

Wer? Florian Tobias Schandinat

Wo? 50.34, Raum -118

Wann? jeden Donnerstag 15:45-17:15

## Material online

[http://github.com/schandinat/algorithmen1\\_ss11](http://github.com/schandinat/algorithmen1_ss11)

## Grundkonzept

Zwischenergebnisse speichern und wiederverwenden

## CYK-Algorithmus

Ist das Wort  $w = bbabaa$  aus  $S$  mittels der folgenden Regeln ableitbar

- $S \rightarrow AB|BC$
- $A \rightarrow BA|a$
- $B \rightarrow CC|b$
- $C \rightarrow AB|a$

## CYK-Algorithmus

Ist das Wort  $w = bbabaa$  aus  $S$  mittels der folgenden Regeln ableitbar

- $S \rightarrow AB|BC$
- $A \rightarrow BA|a$
- $B \rightarrow CC|b$
- $C \rightarrow AB|a$

b	{B}	{ }	{A}	{S,C}	{B}	{A,S}
b	{B}	{S,A}	{S,C}	{B}	{A,S}	
a	{A,C}	{S,C}	{B}	{S,A}		
b	{B}	{A,S}	{ }			
a	{A,C}	{B}				
a	{A,C}					

⇒ ja

<http://de.wikipedia.org/wiki/Cocke-Younger-Kasami-Algorithmus>

## Eigenschaften

- liefern für viele Probleme keine optimale Lösung

## Grundkonzept

Zu jedem Zeitpunkt die Entscheidung treffen, die dann am Optimalsten erscheint

## Beispiele

- Dijkstra-Algorithmus
- Algorithmus von Kruskal
- Heuristik zur Lösung des TSP (nearest neighbour algorithm)

# Übungen

Sortieren Sie 'hans', 'peter', 'fritz', 'horst' und 'anna' alphabetisch aufsteigend mittels Radix-Sort, wobei die Namen jeweils mittels ' ' auf die gleiche Länge aufgefüllt werden und ' ' kleiner als alle Buchstaben sein soll.

Sortieren Sie 'hans', 'peter', 'fritz', 'horst' und 'anna' alphabetisch aufsteigend mittels Radix-Sort, wobei die Namen jeweils mittels ' ' auf die gleiche Länge aufgefüllt werden und ' ' kleiner als alle Buchstaben sein soll.

- Was unterscheidet Radix-Sort von den anderen Sortierv Verfahren, die Sie kennen?

Sortieren Sie 42, 13, 7, 21, 46, 35, 2 mittels Heapsort so dass das Ergebnis eine aufsteigende Reihenfolge hat.



Sortieren Sie 42, 13, 7, 21, 46, 35, 2 mittels Heapsort so dass das Ergebnis eine aufsteigende Reihenfolge hat.

- Was bedeutet inplace?

Es ist die Hashfunktion

$$h(x) = x \bmod 7$$

gegeben. Verwenden Sie offene Adressierung mit linearem Sondieren zur Konfliktauflösung.

- Fügen Sie 0, 7, 1, 14, 6 ein.
- Entfernen Sie die 1.

**Vielen Dank für die  
Aufmerksamkeit!**