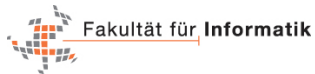


Info IV Tutorium

Thomas Pajor



IBDS Prautzsch

23. April 2007



Euer Tutor

- ▶ Name: Thomas Pajor
- ▶ E-Mail: thomas.pajor@logn.de
- ▶ Tut-Seite: www.logn.de/tut/



Euer Tutor

- ▶ Name: Thomas Pajor
- ▶ E-Mail: thomas.pajor@logn.de
- ▶ Tut-Seite: www.logn.de/tut/

Das Tutorium findet ab jetzt *immer* hier in Raum -109 im Info-Neubau statt!



Vermöge eine Klausur:

- ▶ Hauptklausur ist vor. am **3. August 2007** um 12:00 Uhr
- ▶ Nachklausur ist vor. am **6. Oktober 2007** um 9:00 Uhr

Zu den Übungsblättern und dem Schein:



Vermöge eine Klausur:

- ▶ Hauptklausur ist vor. am **3. August 2007** um 12:00 Uhr
- ▶ Nachklausur ist vor. am **6. Oktober 2007** um 9:00 Uhr

Zu den Übungsblättern und dem Schein:

- ▶ Abgabe dieses Semester nur einzeln bis Mittwoch (Abend)



Vermöge eine Klausur:

- ▶ Hauptklausur ist vor. am **3. August 2007** um 12:00 Uhr
- ▶ Nachklausur ist vor. am **6. Oktober 2007** um 9:00 Uhr

Zu den Übungsblättern und dem Schein:

- ▶ Abgabe dieses Semester nur einzeln bis Mittwoch (Abend)
- ▶ Pro Übungsblatt *eine* Korrekturaufgabe mit 10 Punkten



Vermöge eine Klausur:

- ▶ Hauptklausur ist vor. am **3. August 2007** um 12:00 Uhr
- ▶ Nachklausur ist vor. am **6. Oktober 2007** um 9:00 Uhr

Zu den Übungsblättern und dem Schein:

- ▶ Abgabe dieses Semester nur einzeln bis Mittwoch (Abend)
- ▶ Pro Übungsblatt *eine* Korrekturaufgabe mit 10 Punkten
- ▶ Für den Schein sind $2/3$ der Punkte zu erreichen



Vermöge eine Klausur:

- ▶ Hauptklausur ist vor. am **3. August 2007** um 12:00 Uhr
- ▶ Nachklausur ist vor. am **6. Oktober 2007** um 9:00 Uhr

Zu den Übungsblättern und dem Schein:

- ▶ Abgabe dieses Semester nur einzeln bis Mittwoch (Abend)
- ▶ Pro Übungsblatt *eine* Korrekturaufgabe mit 10 Punkten
- ▶ Für den Schein sind 2/3 der Punkte zu erreichen
- ▶ Dieses Jahr gibt es kein Programmierprojekt!



Algorithmus 1 : NAIVFIND

Eingabe : Text α und Muster β ; $|\alpha| = n$ und $|\beta| = m$

Ausgabe : Erstes Vorkommen von β in α

für $i \leftarrow 1 \dots n - m$ **tue**

┌ **wenn** $\forall j \in \{1, \dots, m\} : \beta_j = \alpha_{i+j-1}$ **dann**
└ ┌ Muster gefunden an Stelle i

Algorithmus 2 : BOYERMOORE

Eingabe : Text α und Muster β

Ausgabe : Erstes Vorkommen von β in α

$\rho \leftarrow \text{BADCHARACTER}(\beta)$

$\sigma \leftarrow \text{STRONGSUFFIX}(\beta)$

für $i \leftarrow 1 \dots n - m$ **tue**

für $j \leftarrow m \dots 1$ **tue**

wenn $\beta_j \neq \alpha_{i+j-1}$ **dann**

$\delta \leftarrow \max\{j - \rho(\alpha_i), j - \sigma(j)\}$

$i \leftarrow i + \delta$

break

 Muster gefunden an Stelle i

Algorithmus 3 : BADCHARACTER

Eingabe : Muster β

Ausgabe : Sprungfunktion $\rho : \Sigma \rightarrow \mathbb{Z}$

für alle $c \in \Sigma$ **tue** */* Alle Werte mit 0 initialisieren */*

└ $\rho(c) \leftarrow 0$

für $j = m \dots 1$ **tue** */* Rechteste Vorkommen in β ermitteln */*

└ **wenn** $\rho(\beta_j) = 0$ **dann**

└└ $\rho(\beta_j) \leftarrow j$

Strong-Suffix-Rule

Algorithmus 4 : STRONGSUFFIX

Eingabe : Muster β , $|\beta| = m$

Ausgabe : Sprungfunktion $\sigma : \{1, \dots, m\} \rightarrow \mathbb{Z}$

// Sonderfall

$\sigma(m) \leftarrow m - 1$

für $j \leftarrow (m - 1) \dots 1$ **tue**

$S \leftarrow \beta_{j+1} \dots \beta_m$

 // Fall I

wenn $k \leftarrow \max\{\kappa < j \mid \beta_{\kappa+1} \dots \beta_{\kappa+m-j} = S \wedge \beta_{\kappa} \neq \beta_j\} \neq \perp$ **dann**

$\sigma(j) \leftarrow k$

 // Fall II

sonst

$k \leftarrow \max(\{\kappa \mid \beta_1 \dots \beta_{\kappa} = \beta_{m-\kappa+1} \dots \beta_m\} \cup \{0\})$

$\sigma(j) \leftarrow k + j - m$



Aufgabe 1

Betrachten Sie den Algorithmus von BOYER & MOORE.

- (a) Berechnen Sie die Sprungtabellen ρ und σ für das Muster MISSISSIPPI unter Verwendung von $\Sigma := \{I, M, N, P, S\}$.
- (b) Führen Sie den Algorithmus am Beispiel

`$\alpha :=$ this is a simple example`

mit dem Muster `example` durch, wobei

$$\Sigma := \bigcup_{i=1}^{|\alpha|} \{\alpha_i\}.$$