UA RUHR

Algorithmen auf Sequenzen

Exakte Mustersuche

Sven Rahmann

Genominformatik
Universitätsklinikum Essen
Universität Duisburg-Essen
Universitätsallianz Ruhr

Das Pattern Matching Problem (Exakte Mustersuche)

UA RUHR

Gegeben sei ein endliches Alphabet Σ, ein Text $T \in \Sigma^n$ und ein Muster (Pattern) $P \in \Sigma^m$ i.d.R. $m \ll n$.

Gesucht sind alle Positionen i, für die gilt: $T[i : i + m] = P$.

Das Pattern Matching Problem (Exakte Mustersuche)

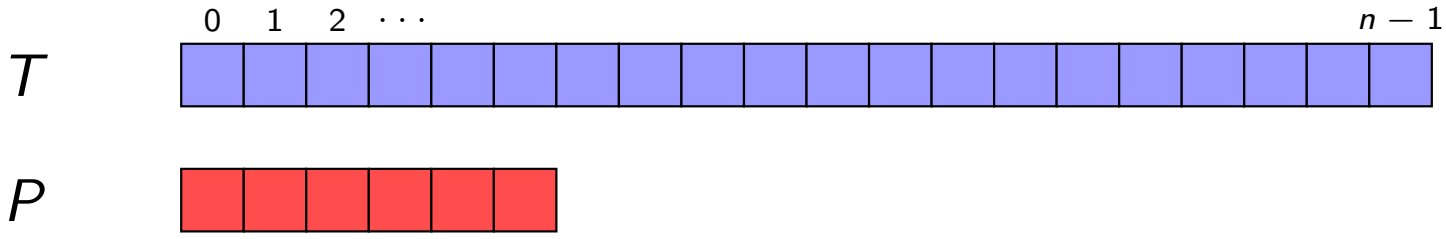
UA RUHR

Gegeben sei ein endliches Alphabet Σ, ein Text $T \in \Sigma^n$ und ein Muster (Pattern) $P \in \Sigma^m$ i.d.R. $m \ll n$.

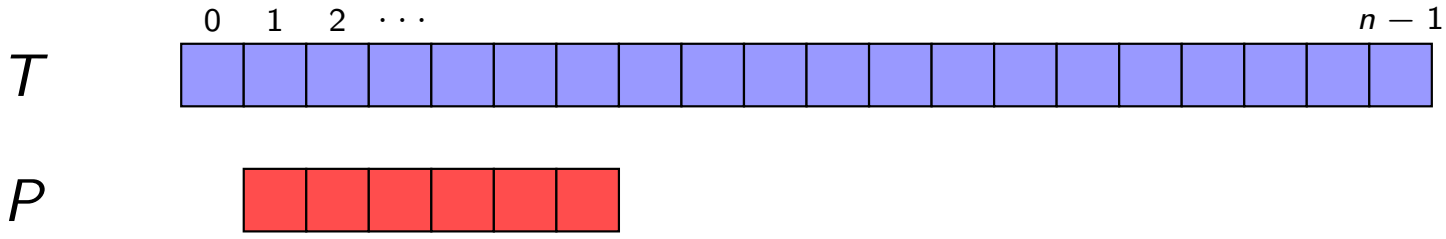
Gesucht sind alle Positionen i, für die gilt: $T[i : i + m] = P$.
Varianten:

- Kommt P in T vor?
- Wie oft kommt P in T vor?

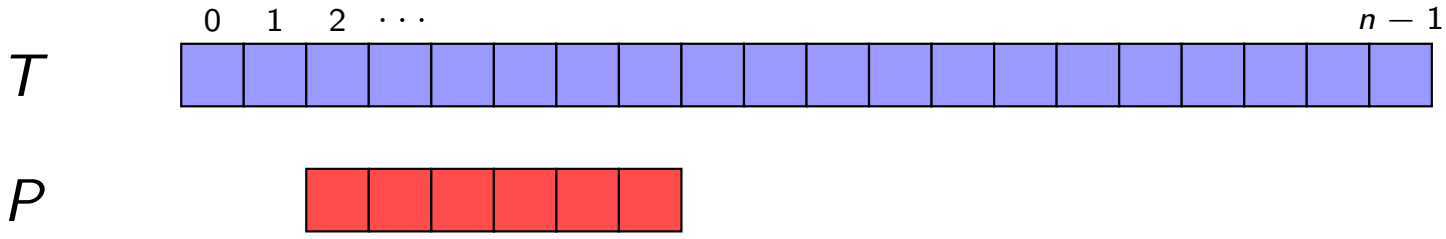
Naiver Algorithmus



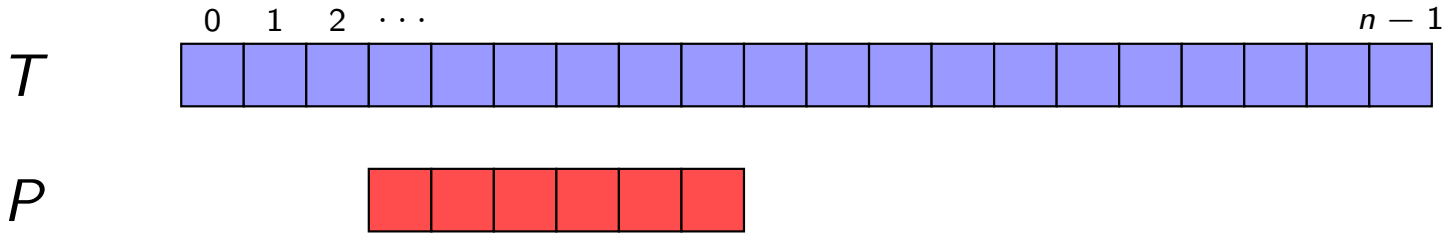
Naiver Algorithmus



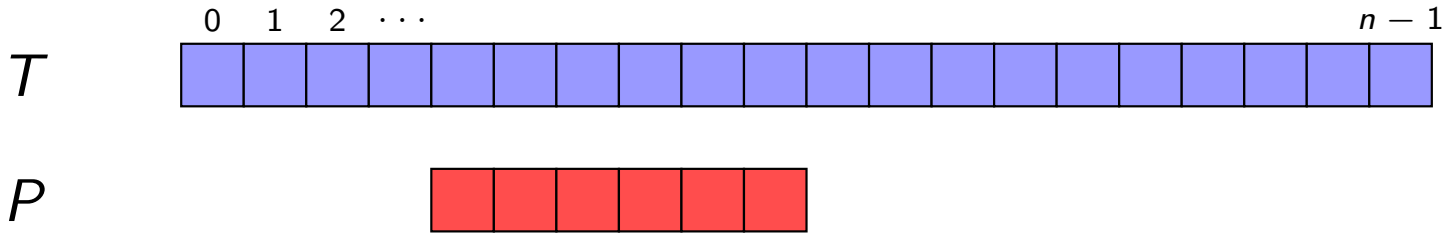
Naiver Algorithmus



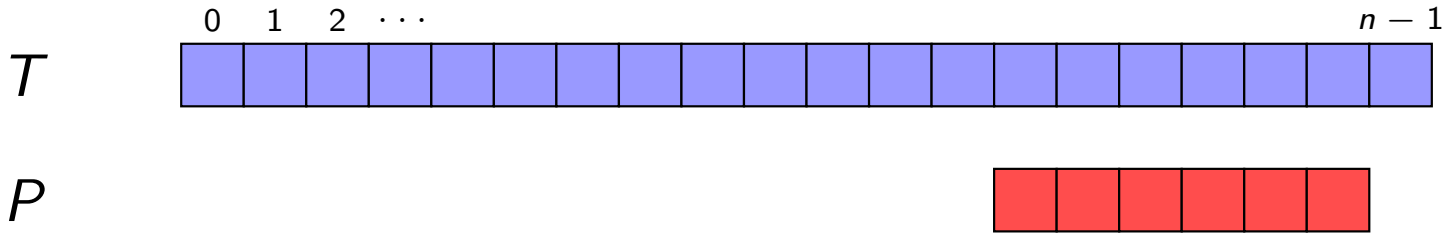
Naiver Algorithmus



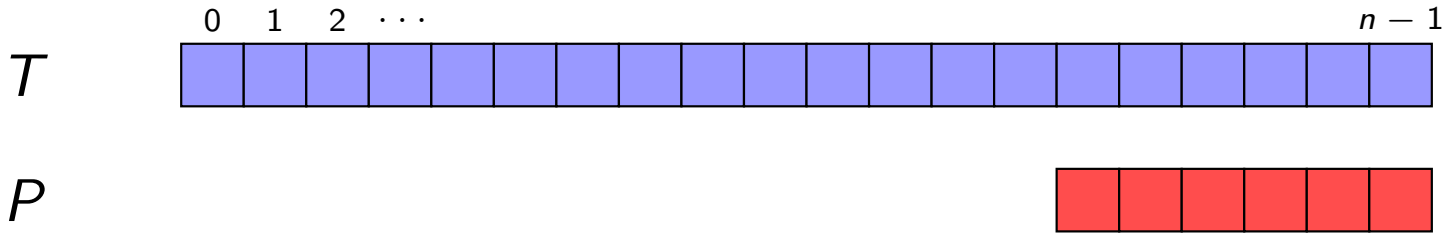
Naiver Algorithmus



Naiver Algorithmus

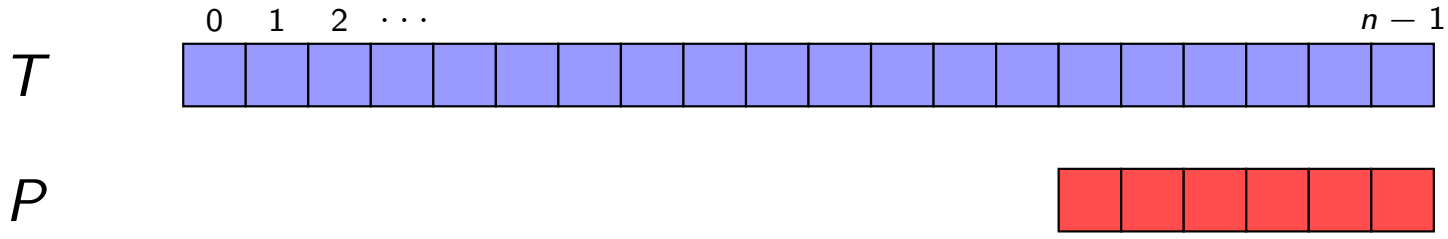


Naiver Algorithmus



Insgesamt $n - m + 1$ Verschiebungen.
Pro Verschiebung bis zu m Vergleiche.
Gesamte Laufzeit: $\mathcal{O}(mn)$.

Naiver Algorithmus



Insgesamt $n - m + 1$ Verschiebungen.
Pro Verschiebung bis zu m Vergleiche.
Gesamte Laufzeit: $\mathcal{O}(mn)$.

```
1 def naive(P, T):
2     m, n = len(P), len(T)
3     for i in range(n - m + 1):
4         if T[i:i+m] == P:
5             yield (i, i+m)
```

Laufzeitanalyse des naiven Algorithmus

UA RUHR

Theorem (Erwartete Laufzeit)

Sei $|\Sigma| \geq 2$. Seien ein Muster der Länge m und ein Text der Länge n zufällig gleichverteilt gewählt. Dann beträgt die Worst-case-Laufzeit des naiven Algorithmus $\mathcal{O}(mn)$, aber die erwartete Laufzeit lediglich $\mathcal{O}(n)$.

Laufzeitanalyse des naiven Algorithmus

Die Wahrscheinlichkeit p, dass jeweils ein zufällig gezogenes Zeichen aus P und T übereinstimmen, beträgt

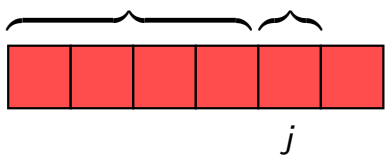
$$p := \frac{|\Sigma|}{|\Sigma|^2} = \frac{1}{|\Sigma|}.$$

Laufzeitanalyse des naiven Algorithmus

Die Wahrscheinlichkeit p, dass jeweils ein zufällig gezogenes Zeichen aus P und T übereinstimmen, beträgt

$$p := \frac{|\Sigma|}{|\Sigma|^2} = \frac{1}{|\Sigma|}.$$

- Die W'keit, dass alle Zeichen des Musters mit dem Text übereinstimmen, beträgt p^m.
- Die W'keit, dass das Muster erst an Stelle j mit dem Text nicht übereinstimmt, beträgt $p^{j-1} \cdot (1-p)$.



Laufzeitanalyse des naiven Algorithmus

Die erwartete Anzahl an Vergleichen für ein Muster der Länge m beträgt demnach:

$$E_m := mp^m + \sum_{j=1}^{m} jp^{j-1} \cdot (1-p)$$

Laufzeitanalyse des naiven Algorithmus

Die erwartete Anzahl an Vergleichen für ein Muster der Länge m beträgt demnach:

$$E_m := mp^m + \sum_{j=1}^{m} jp^{j-1} \cdot (1-p)$$

Und für beliebige Längen $m \to \infty$:

$$E_m < E_\infty := (1-p) \sum_{j=0}^{\infty} jp^{j-1}$$

Laufzeitanalyse des naiven Algorithmus

Der Term $\sum_{j=0}^{\infty} jp^{j-1}$ ist die Ableitung von $\sum_{j=0}^{\infty} p^j = 1/(1-p)$.
Somit gilt $\sum_{j=0}^{\infty} jp^{j-1} = 1/(1-p)^2$.

Laufzeitanalyse des naiven Algorithmus

Der Term $\sum_{j=0}^{\infty} jp^{j-1}$ ist die Ableitung von $\sum_{j=0}^{\infty} p^j = 1/(1-p)$.
Somit gilt $\sum_{j=0}^{\infty} jp^{j-1} = 1/(1-p)^2$.

Eingesetzt in E_∞ ergibt es

$$E_\infty = \frac{1-p}{(1-p)^2} = \frac{1}{1-p}.$$

Laufzeitanalyse des naiven Algorithmus

Der Term $\sum_{j=0}^{\infty} jp^{j-1}$ ist die Ableitung von $\sum_{j=0}^{\infty} p^j = 1/(1-p)$.
Somit gilt $\sum_{j=0}^{\infty} jp^{j-1} = 1/(1-p)^2$.

Eingesetzt in E_∞ ergibt es

$$E_\infty = \frac{1-p}{(1-p)^2} = \frac{1}{1-p}.$$

Aus der Definition $p = 1/|\Sigma|$ folgt nun

$$E_m < \frac{|\Sigma|}{|\Sigma| - 1}.$$

Laufzeitanalyse des naiven Algorithmus

Der Term $\sum_{j=0}^{\infty} jp^{j-1}$ ist die Ableitung von $\sum_{j=0}^{\infty} p^j = 1/(1-p)$.
Somit gilt $\sum_{j=0}^{\infty} jp^{j-1} = 1/(1-p)^2$.

Eingesetzt in E_∞ ergibt es

$$E_\infty = \frac{1-p}{(1-p)^2} = \frac{1}{1-p}.$$

Aus der Definition $p = 1/|\Sigma|$ folgt nun

$$E_m < \frac{|\Sigma|}{|\Sigma|-1}.$$

Für ein 2-buchstabiges Alphabet beträgt $E_m < 2$, für $|\Sigma| \to \infty$ sogar nur $E_m = 1$. Die Laufzeit entspricht somit $\mathcal{O}(nE_m) = \mathcal{O}(n)$.

□