

---

## Algorithmische Bioinformatik I

---

### Aufgabe 1

Wir betrachten zunächst die einfache Variante vom AC-Algorithmus, bei der kein Suchwort in einem anderen Suchwort enthalten sein darf. Bisher wurde nur ausgegeben *ob irgendein* Suchwort im Text enthalten ist. Nun soll außerdem auch ausgegeben werden, *welches Suchwort an welcher Position* gefunden wurde. Beschreiben Sie in Worten, wie der Algorithmus und die Datenstruktur angepasst werden müssen.

Wie funktioniert es, wenn zugelassen wird, dass Suchwörter in anderen Suchwörtern enthalten sind?

### Aufgabe 2

Bestimme für die Zeichenreihe

$$s = \text{abaabbabab}$$

die Shift-Tabelle für den Boyer-Moore-Algorithmus.

### Aufgabe 3

Betrachte den Boyer-Moore Algorithmus, der für die Bestimmung der Shifts bei einem Mismatch nur die Bad-Character-Rule berücksichtigt:

```
1 bmc_bad_character(char t[], int n, char s[], int m) {
2   int i := 0;
3   int j := m - 1;
4   while (i ≤ n - m) {
5     while (t[i+j] == s[j]) {
6       if (j == 0) return true;
7       j--;
8     }
9     i := i + j - max{k | (k < j ∧ s[k] = t[i + j]) ∨ (k = -1)};
10    j := m - 1;
11  }
12  return false;
13 }
```

Man gebe eine unendliche Familie von Beispielen an, bei dem diese Variante eine Laufzeit von  $\Theta(n \cdot m)$  hat.

### Aufgabe 4

Man gebe für jedes Paar  $n, m \in \mathbb{N}, m, n \geq 2$  jeweils einen Text  $t$  der Länge  $n$  und ein Suchmuster  $s$  der Länge  $m$  an, so dass der Algorithmus von Knuth, Morris und Pratt mindestens  $2n - m$  Vergleiche ausführt (die Vergleiche zum Erstellen der Tabelle  $border[]$  sind hierbei nicht zu berücksichtigen).

## Aufgabe 5

Ein Wort  $w \in \Sigma^*$  heißt *periodisch*, wenn es ein Wort  $v \in \Sigma^*$  und ein  $i \in \mathbb{N}$  gibt, so dass  $|v| \geq 1$ ,  $n > 1$  und  $w = v^i$ .

Konstruiere einen Algorithmus an, der mit linear vielen Vergleichen feststellt, ob ein Wort periodisch ist.