

3. Übung zur Vorlesung “Einführung in die Bioinformatik I, 1. Teil”

Wintersemester 2018/2019

Prof. Sebastian Böcker, Emanuel Barth, Maximilian Collatz, Marcus Ludwig

Ausgabe: 07. November 2018,
Abgabe: 14. November 2018 zu Beginn der Übung

Aufgabe 1 (5 Punkte):

1. Denken Sie sich je einen Text der Länge zehn und ein Pattern der Länge vier aus, sodass das Pattern genau einmal im Text vorkommt und der naive Textsuche-Algorithmus (a) möglichst viele und (b) möglichst wenige Schritte benötigt. Wie viele Schritte sind es?
2. Lösen Sie die gleiche Aufgabe für Texte der Länge n und Patterns der Länge m , und geben Sie die Zahl der jeweils benötigten Schritte an.

Aufgabe 2 (5 Punkte):

1. Erstellen Sie Z-Arrays für die folgenden Strings: `abrakadabra`, `aabaacaabaabaacaa` und `tataatataataaaaa`.
2. Zeigen Sie, wie der lineare Algorithmus mit Z-Vorverarbeitung im Text `Mississippi` nach Pattern `issip` und im Text `abcdeaabcdababcababcababcdaabcde` nach Pattern `abc` sucht.

Aufgabe 3 (5 Punkte):

1. Schätzen Sie die folgenden Ausdrücke möglichst einfach mit Hilfe der O-Notation ab, z. B. $n^2 + 1 = O(n^2)$.
 $5n^3 + 6n$, $3n^2 + \frac{n^3}{4}$, $6n^4 + 7 \cdot 2^n$, $(n + 3)(n - 2)$, $n + \log n$
2. Vergleichen Sie mit Hilfe von Landausymbolen (Ω , Θ , O) jeweils zwei Funktionen miteinander: n mit $2n$, $\log n$ mit $\log n^2$, 2^n mit 2^{n+1} , 2^n mit 2^{2n} , $(n + 7)^5$ mit n^5 . Wählen Sie, falls möglich, das Symbol mit der stärksten Aussagekraft.

(bitte wenden)

Aufgabe 4 (5 Punkte): Betrachten Sie die folgenden Algorithmen zum Sortieren eines eingegebenen Arrays:

```
sort_1 (ARRAY a)
  FOR i ← 1 TO length(a) DO
    min ← ∞;
    FOR j ← i TO length(a) DO
      IF a[j] < min THEN
        min ← a[j];
        minindex ← j;
      END IF;
    END FOR;
    swap(a[i], a[minindex]);
  END FOR;
  RETURN a;
END.

sort_2 (ARRAY a)
  FOR i ← 1 TO length(a) DO
    FOR j ← length(a) DOWNTO i+1 DO
      IF a[j] < a[j-1] THEN
        swap(a[j], a[j-1]);
      END IF;
    END FOR;
  END FOR;
  RETURN a;
END.
```

Dabei ist $a[i]$ das i -te Element und $\text{length}(a)$ die Länge des Arrays a .

1. Demonstrieren Sie anhand der Eingaben (3, 1, 2, 4) und (5, 4, 1, 3, 2) die Funktionsweise der Algorithmen.
2. Geben Sie mittels O-Notation eine möglichst kleine obere Schranke für die Laufzeiten der beiden Algorithmen an!