

Aufgabe 1

Beschreibe, was ein *Suchverfahren* ist und gib einige Beispiele dafür an.

Aufgabe 2

Welche Laufzeitkomplexitäten haben die folgenden Suchalgorithmen?

- (a) Suche in einer unsortierten Liste
- (b) Suche in einer sortierten Liste
- (c) Tiefensuche in einem Graphen mit $|V|$ Knoten und $|E|$ Kanten in Adjazenzlisten-darstellung
- (d) Breitensuch in einem Graphen mit $|V|$ Knoten und $|E|$ Kanten in Adjazenzlisten-darstellung
- (e) Naive Suche nach einer Zeichenkette mit m Zeichen in einer Zeichenkette mit n Zeichen

Aufgabe 3

- (a) Schreibe eine Python-Funktion `is_element_of(L, item)`, deren Parameter eine Liste `L` und ein beliebiges Objekt `item` sind und die `True` zurückgibt, wenn sich das Objekt in der Liste befindet und `False` sonst.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|
| 1 | d | e | f | | i | s | _ | e | l | e | m | e | n | t | _ | o | f | (| L | , | i | t | e | m |) | : | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | |

- (b) Welche Laufzeit(en) hat diese Funktion?
- (c) Beschreibe, wie man die Funktion verändern müsste, wenn sie eine Liste der Indizes zurückgeben soll, an denen sich das Element befindet.

Aufgabe 5

- (a) Bestimme die Anzahl der Vergleiche, die der „naive“ Algorithmus für das String-Matching zum Auffinden des Musters **GGCA** im Textstring **GGGAAAGGCAT** benötigt.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|
| G | G | G | A | A | A | G | G | C | A | T | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

- (b) Leite die Worst-Case-Laufzeit für die Suche nach einem Muster p der Länge m in einem Text t der Länge n anhand eines geeigneten Beispiels her.

Aufgabe 6

Bestimme die Anzahl der Vergleiche, die der Boyer-Moore-Horspool-Algorithmus für das String-Matching zum Auffinden des Musters GGCA im Textstring GGGAAAGGCAT benötigt.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|
| G | G | G | A | A | A | G | G | C | A | T | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

Aufgabe 7

Erstelle schrittweise die Bad-Character-Table des Boyer-Moore-Horspool-Algorithmus für das Suchmuster SALATELLER. Zeichen des Alphabets, die nicht im Suchmuster vorkommen, sind durch einen Stern (*) darzustellen.

| | | | | | | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

