

Suchalgorithmen

Übungen

Aufgabe 1

Um ein Element x_1 in einem Array A_1 der Länge 1000 zu finden, benötigt ein sequentieller Algorithmus etwa 2 Sekunden. Wie lange benötigt derselbe Algorithmus, um ein Element x_2 in einem Array A_2 der Länge 2000 zu finden?

- (a) etwa 4 Sekunden
- (b) etwa 2 Sekunden
- (c) keine Vorhersage möglich

Aufgabe 1

Da die 2 Sekunden davon abhängig sind, ob und wo sich das Element x_1 im Array A_1 befindet, lässt sich aus dem selben Grund für ein möglicherweise anderes Element x_2 in einer möglicherweise anderen Array A_2 keine Aussage machen. Also stimmt (c).

Aufgabe 2

Gib die Laufzeitkomplexität der sequentiellen Suche in einem Array der Länge n für die folgenden Fälle an:

- (a) Best Case
- (b) Average Case
- (c) Worst Case

Aufgabe 2

Es müssen zwei Fälle unterschieden werden:

- (a) Das Element befindet sich in der Liste:

Best Case: $O(1)$

Average Case: $O(n/2) = O(n)$

Worst Case: $O(n)$

- (b) Das gesuchte Element befindet sich nicht in der Liste:

Best Case: $O(n)$

Average Case: $O(n)$

Worst Case: $O(n)$

Aufgabe 3

Eine Implementierung des Algorithmus für binäres Suchen benötigt 2 Sekunden, um herauszufinden, dass ein Element *nicht* in einem Array mit 10^4 Elementen vorkommt.

Wie lange benötigt dieselbe Implementierung auf dem gleichen Computer um herauszufinden, dass sich ein Element *nicht* in einem Array mit 10^8 Elementen befindet?

Aufgabe 3

Laufzeitkomplexität für binäres Suchen: $T(n) = c \cdot \log n$

Wähle hier die Logarithmenbasis 10 (bequem zum Rechnen)

$$T(10^4) = C \log_{10}(10^4) = 4C = 2\text{ s} \quad \Rightarrow \quad C = \frac{1}{2}\text{ s}$$

$$T(10^8) = C \log_{10}(10^8) = \frac{1}{2}\text{ s} \cdot 8 = 4\text{ s}$$

oder direkt durch Umformung:

$$T(10^8) = C \log_a(10^8) = C \log_a(10^4)^2 = 2 \cdot C \log_a(10^4) = 2 \cdot 2\text{ s} = 4\text{ s}$$

Aufgabe 4

Die Zeitmessung für eine Implementierung des Algorithmus für binäres Suchen hat für die erfolglose Suche eines Elements in einem Array mit 10^6 Elementen eine Laufzeit von 20 Sekunden ergeben.

Wie lange benötigt dieselbe Implementierung auf dem gleichen Computer, um herauszufinden, dass sich das Element *nicht* in einem Array mit jeweils $2 \cdot 10^6$ Elementen befindet?

Aufgabe 4

$$T(n) = C \cdot \log_b n$$

Wähle als Logarithmenbasis 2

$$\begin{aligned} T(10^6) &= C \cdot \log_2 10^6 = C \cdot \log_2 (10^3)^2 = 2C \cdot \log_2 10^3 \\ &\approx 2C \cdot \log_2 2^{10} = 20C = 20 \text{ s} \end{aligned}$$

$$\Rightarrow C = 1 \text{ s}$$

Aufgabe 4

$$T(n) = C \cdot \log_b n$$

Wähle als Logarithmenbasis 2

$$\begin{aligned} T(10^6) &= C \cdot \log_2 10^6 = C \cdot \log_2 (10^3)^2 = 2C \cdot \log_2 10^3 \\ &\approx 2C \cdot \log_2 2^{10} = 20C = 20 \text{ s} \end{aligned}$$

$$\Rightarrow C = 1 \text{ s}$$

$$\begin{aligned} T(2 \cdot 10^6) &= C \cdot (\log_2 2 + \log_2 10^6) = C \cdot 1 + C \cdot \log_2 10^6 \\ &= 1 \text{ s} + 20 \text{ s} = 21 \text{ s} \end{aligned}$$

Aufgabe 5

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

Aufgabe 5

G	G	G	A	A	A	G	G	C	A	T	Vergleiche
G	G	C	A								3
	G	G	C	A							3
		G	G	C	A						2
			G	G	C	A					1
				G	G	C	A				1
					G	G	C	A			1
						G	G	C	A		4
Total											15

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.

Aufgabe 6

pattern=GGCA ($m = 4$)

Bad Character Table:

Character	G	C	A	*
Shift	2	1	4	4

Das Symbol * steht für alle Buchstaben, die nicht im Suchmuster vorkommen.

Shift = Wert[pattern[j]] = $m-j-1$ ($j=0, \dots, m-2$)

G	G	G	A	A	A	G	G	C	A	T	Vergleiche
G	G	C	A								2
				G	G	C	A				1
						G	G	C	A		4
Total											7

Aufgabe 7

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

Aufgabe 7

Suchmuster: SALATTeller (Länge: 11)

S	A	L	T	E	R	*	
11	11	11	11	11	11	11	Initialisierung
10	11	11	11	11	11	11	$d(\text{S}, \text{R}) = 11 - 1 = 10$
10	9	11	11	11	11	11	$d(\text{A}, \text{R}) = 11 - 2 = 9$
10	9	8	11	11	11	11	$d(\text{L}, \text{R}) = 11 - 3 = 8$
10	7	8	11	11	11	11	$d(\text{A}, \text{R}) = 11 - 4 = 7$
10	7	8	6	11	11	11	$d(\text{T}, \text{R}) = 11 - 5 = 6$
10	7	8	5	11	11	11	$d(\text{T}, \text{R}) = 11 - 6 = 5$
10	7	8	5	4	11	11	$d(\text{E}, \text{R}) = 11 - 7 = 4$
10	7	3	5	4	11	11	$d(\text{L}, \text{R}) = 11 - 8 = 3$
10	7	2	5	4	11	11	$d(\text{L}, \text{R}) = 11 - 9 = 2$
10	7	2	5	1	11	11	$d(\text{E}, \text{R}) = 11 - 10 = 1$

Aufgabe 8

Gib den deterministischen endlichen Automaten an (graphisch oder als Tabelle), mit dem in einem beliebigen Text nach dem Muster ABAB gesucht werden kann.

Aufgabe 8

s ist das längste Suffix von $p[0:i]+x$, das Präfix des Suchmusters $p = \text{ABAB}$ ist.

q_i	$p[0:i]+x$	s	q_{i+1}
0	A	A	1
0	B	ε	0
1	AA	A	1
1	AB	AB	2
2	ABA	ABA	3
2	ABB	ε	0
3	ABAA	A	1
3	ABAB	ABAB	4
4	ABABA	ABA	3
4	ABABB	ε	0

Der DFA in graphischer Darstellung:

