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

(a) Best Case: $O(1)$
Das gesuchte Element steht an erster Stelle.

(b) Average Case: $O(n)$
Ist die Chance, dass sich ein Elemente an einer bestimmten Position befindet, überall gleich gross, dann befindet sich das gesuchte Element im Durchschnitt in der „Mitte“ der Liste. Somit gilt: $O(n/2) = O(n)$.

(c) Worst Case: $O(n)$
Das gesuchte Element kommt am Ende des Arrays oder gar nicht in ihm vor.

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)

$$T(10^4) = C \log_{10}(10^4) = 4C = 2\,\mathrm{s} \quad \Rightarrow \quad C = \tfrac{1}{2}\,\mathrm{s}$$

$$T(10^8) = C \log_{10}(10^8) = \tfrac{1}{2}\,\mathrm{s} \cdot 8 = 4\,\mathrm{s}$$

oder direkt durch Umformung:

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

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

$$T(10^6) = C \cdot \log_2 10^6 = C \cdot \log_2 \left(10^3\right)^2 = 2C \cdot \log_2 10^3$$
$$\approx 2C \cdot \log_2 2^{10} = 20C = 20\,\mathrm{s}$$

$$\Rightarrow \quad C = 1\,\mathrm{s}$$

Aufgabe 4

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

Wähle als Logarithmenbasis 2

$$T(10^6) = C \cdot \log_2 10^6 = C \cdot \log_2 \left(10^3\right)^2 = 2C \cdot \log_2 10^3$$
$$\approx 2C \cdot \log_2 2^{10} = 20C = 20\,\mathrm{s}$$

$$\Rightarrow \quad C = 1\,\mathrm{s}$$

$$T(2 \cdot 10^6) = C \cdot \left(\log_2 2 + \log_2 10^6\right) = C \cdot 1 + C \cdot \log_2 10^6$$
$$= 1\,\mathrm{s} + 20\,\mathrm{s} = 21\,\mathrm{s}$$

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, ..., 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 die Boyer-Moore-Bad-Character-Tabelle für das Suchmuster `SALATTELLER`. Zeichen des Alphabets, die nicht im Suchmuster vorkommen, sind mit einem Stern (*) zu berücksichtigen.

Aufgabe 7

Suchmuster: SALATTELLER (Länge: 11)

S	A	L	T	E	R	*
10	7	2	5	1	11	11

Aufgabe 8

Gib den determinstischen 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 = 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: