Technische Universität Braunschweig

Algorithmen und Datenstrukturen – Übung #8

Fragestunde

Matthias Konitzny, Arne Schmidt

11.02.2021

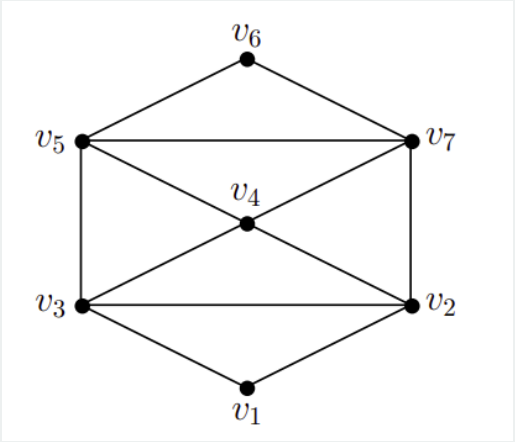
Quiz!

Prüfung

Testlauf

1 Graphenscan - Anwenden

Betrachte folgenden Graphen.



Führe Breitensuche auf diesem Graphen aus. Starte bei Knoten v_1. Kommen in einem Schritt des Algorithmus mehrere Knoten in Frage, wähle denjenigen mit dem kleinsten Index.

1.1	Welche der folgenden Kanten werden in den Breitensuchbaum aufgenommen? (Teil 1)	☐ $\{v_1,v_2\}$ ☐ $\{v_2,v_3\}$	☐ $\{v_1,v_3\}$
1.2	Welche der folgenden Kanten werden in den Breitensuchbaum aufgenommen? (Teil 2)	☐ $\{v_2,v_4\}$ ☐ $\{v_4,v_7\}$	☐ $\{v_2,v_7\}$
1.3	Welche der folgenden Kanten werden in den Breitensuchbaum aufgenommen? (Teil 3)	☐ $\{v_3,v_4\}$ ☐ $\{v_4,v_5\}$	☐ $\{v_3,v_5\}$
1.4	Welche der folgenden Kanten werden in den Breitensuchbaum aufgenommen? (Teil 4)	☐ $\{v_5,v_6\}$ ☐ $\{v_6,v_7\}$	☐ $\{v_5,v_7\}$

Testlauf

2 Master-Theorem

Betrachte folgende Rekursionsgleichung.

$$T(n) := 3 \cdot T\left(\frac{n}{3}\right) - 3 + 2n$$

2.1	Gib alle im Master-Theorem auftretenden Parameter an.	m = [ ] , k = [ ] , α1 = ... = α3 = [ ] / [ ]
2.2	Die Summe der α_i hoch k ist...	○ < 1 ○ = 1 ○ > 1
2.3	Mit dem Master-Theorem ergibt sich somit welche Laufzeit?	○ Θ(n) ○ Θ(n log n) ○ $\Theta(n^2)$

3 Algorithmenentwurf

3.1	Beschreibe kurz wie aus einem gegebenen sortierten Array ein AVL-Baum in O(n) Zeit konstruiert werden kann.	

Einladungsmail an die tubs-Adresse spätestens morgen an alle, die für die Prüfung angemeldet sind.

Fragerunde

Themen

- AVL-Bäume: Rotationen
- Hierholzer: Angabe von Wegen
- Wachstum von Funktionen
  - Vergleichen von Klassen
  - Bestimmen der Klassen
- Algorithmenentwurf
- Bestimmung der Laufzeit von Algorithmen

AVL-Bäume Rotation

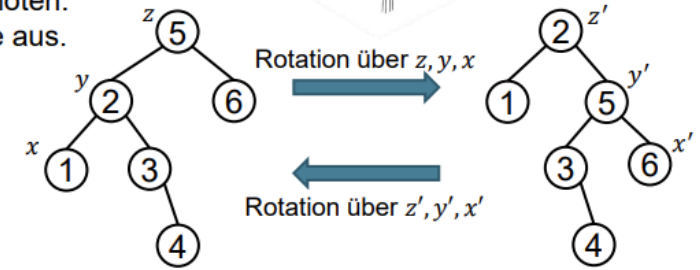
AVL-Bäume – Restructure

Bei Insert und Delete stellen sich nun folgende Fragen:

1. Welche Knoten werden unbalanciert?
2. Wie stellt man die Balance wieder her?
3. Welche Regeln sollte man berücksichtigen?

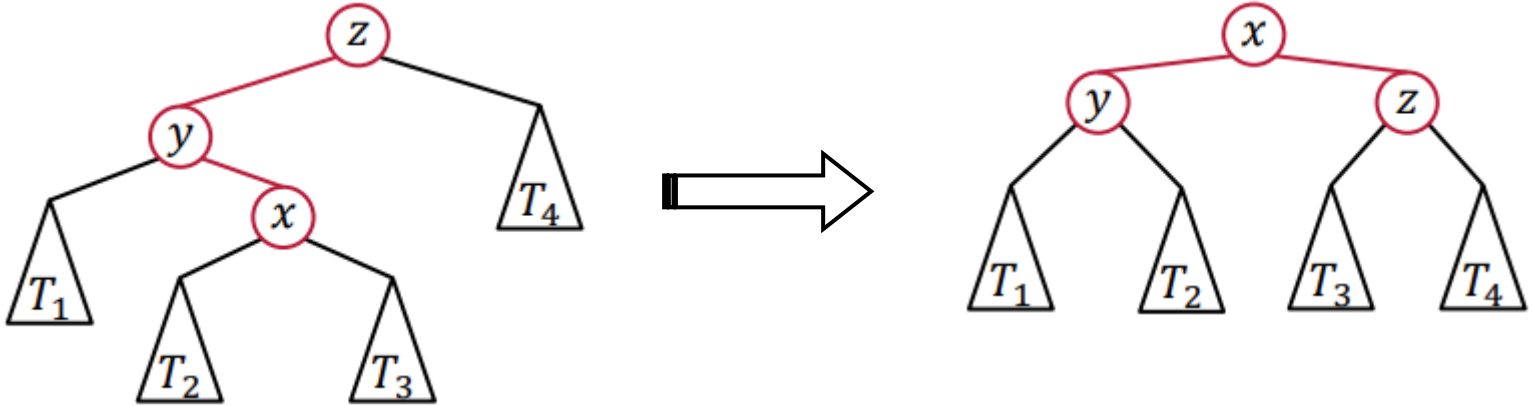
Zu 3.:

- Starte bei tiefstem unbalancierten Knoten.
- Wähle Kinder (x, y) nach deren Höhe aus.

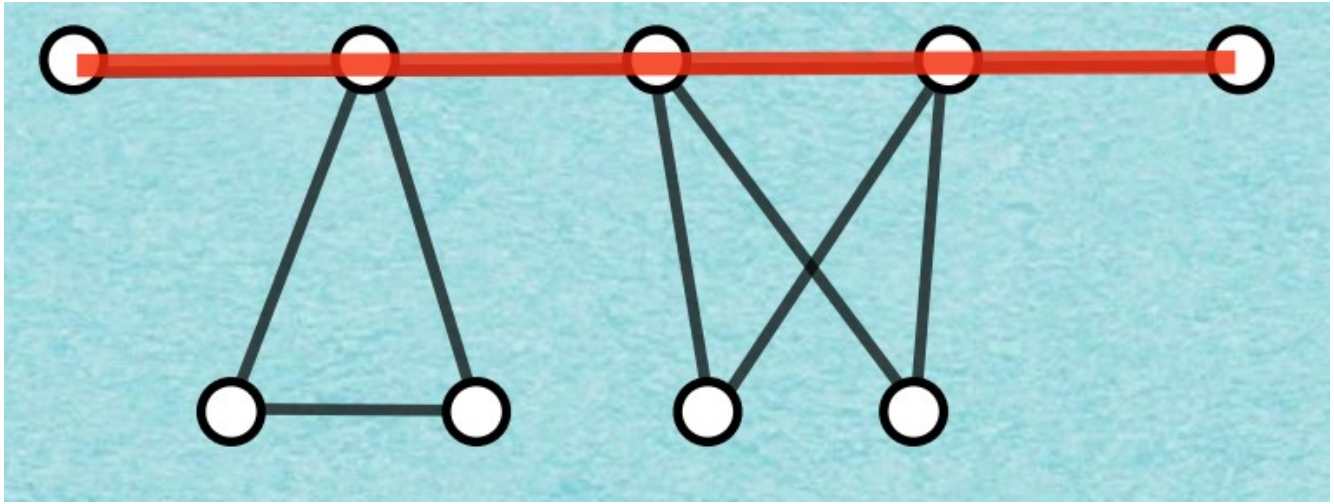


Übung 5

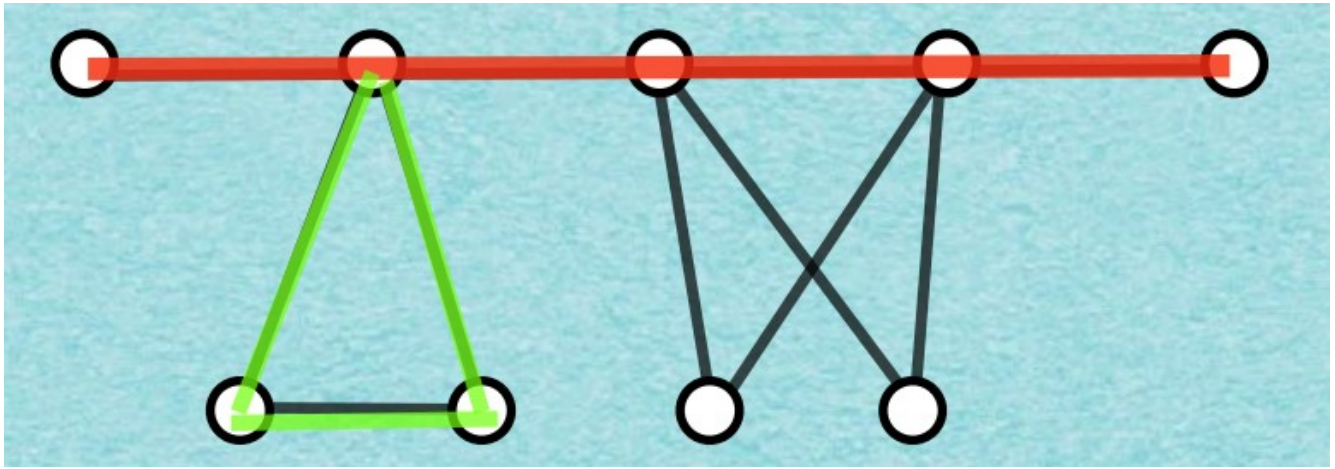
AVL-Bäume Rotation



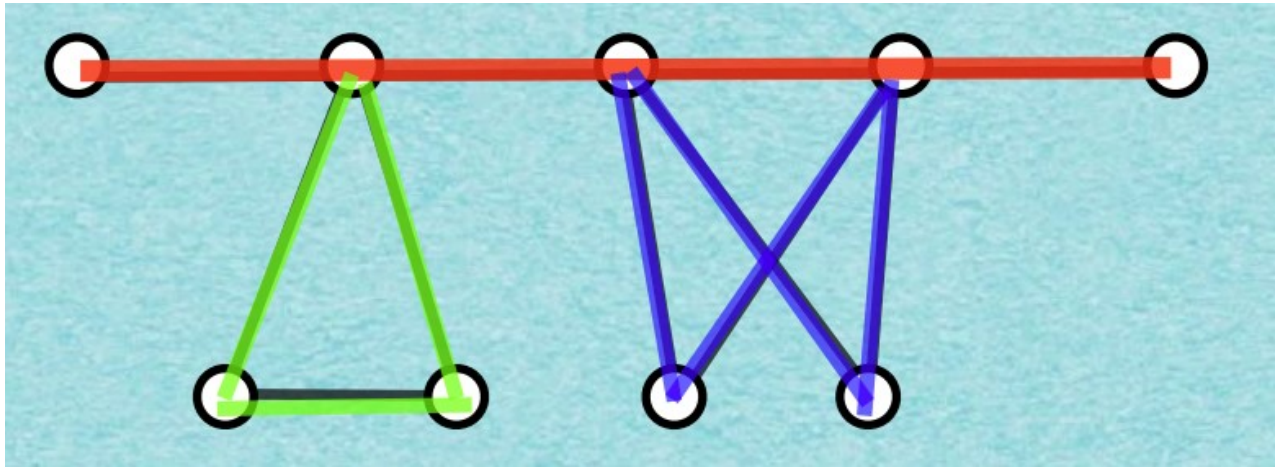
Hierholzer – Eulertouren finden



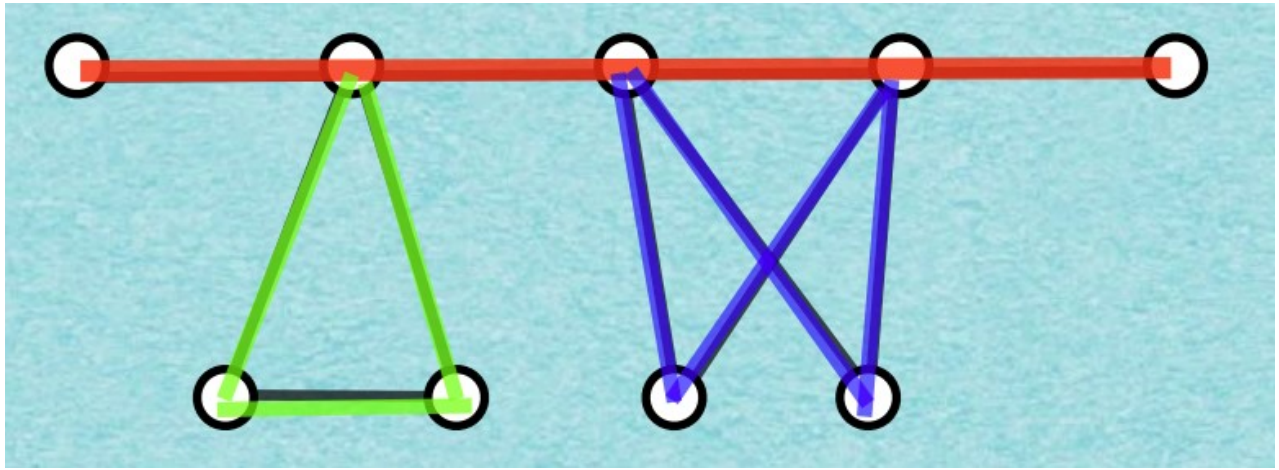
Hierholzer – Eulertouren finden



Hierholzer – Eulertouren finden



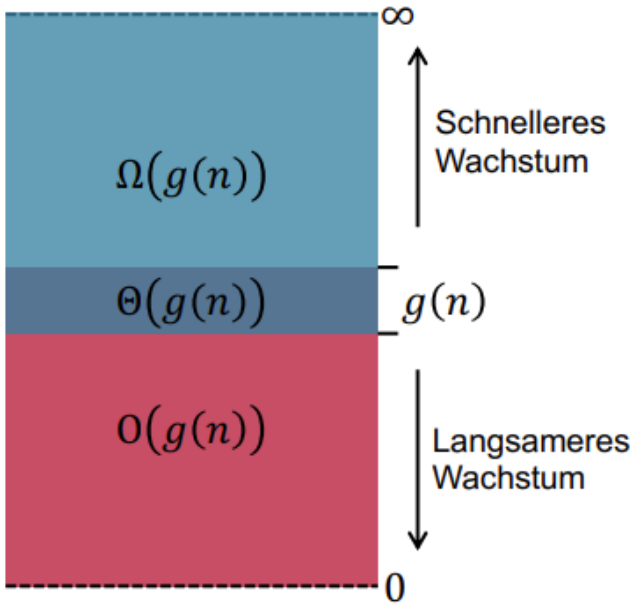
Hierholzer – Eulertouren finden



In der Klausur können alle drei Kantenzüge separat angegeben werden.
Zusätzlich: Stelle markieren, wo welcher Kantenzug eingefügt wird.

Wachstum von Funktionen

Vergleichen von Klassen



Hierarchie-Ausschnitt:

$$O(1) \subset O(\log^a n) \subset O(n^b) \subset O(c^n) \subset O(n!) \subset O(n^n)$$

Bei Ω dreht sich das Inklusionszeichen um!

Wo passt dort nun $O(n \log n)$ rein?

Wie steht das zu $O(n \log \log n)$?

Wir wissen:

$$O(\log \log n) \subset O(\log n)$$

Also muss gelten:

$$O(n \log \log n) \subset O(n \log n)$$

Wachstum von Funktionen

Bestimmen von Klassen

Laufzeiten Merkzettel

1 Definitionen

O-Notation Gibt eine obere Schranke für Funktionen. Gilt $f(n) \in O(g(n))$, so wächst $f(n)$ (asymptotisch) nicht schneller als $g(n)$ denn:

Es existieren zwei Konstanten $c \in \mathbb{R}^+$ und $n_0 \in \mathbb{N}$, sodass für alle $n \geq n_0$ die Ungleichung $0 \leq f(n) \leq c \cdot g(n)$ gilt.

Ω-Notation Gibt eine untere Schranke für Funktionen. Gilt $f(n) \in \Omega(g(n))$, so wächst $f(n)$ (asymptotisch) nicht langsamer als $g(n)$ denn:

Es existieren zwei Konstanten $c \in \mathbb{R}^+$ und $n_0 \in \mathbb{N}$, sodass für alle $n \geq n_0$ die Ungleichung

2 O-Notation

Tipps zum Abschätzen von Funktionen bei der O-Notation:

1. Bei Polynomen können Subtrahenden einfach ignoriert werden. Das Weglassen macht die Funktion nur größer.
2. Bei Polynomen können alle Exponenten von (positiven) Summanden auf den Grad des Polynoms hochgestuft werden. Das macht die Funktion größer.
3. Bei Funktionen die kein Polynom sind, können andere Methoden zum Abschätzen vorteilhaft sein, z.B. das Benutzen von monoton-wachsenden Funktionen (Logarithmieren, Potenzieren[1], Wurzelziehen, etc.).

3 Ω-Notation

Tipps zum Abschätzen von Funktionen bei der Ω-Notation:

1. Bei Polynomen können (positive) Summanden einfach ignoriert werden. Das Weglassen macht die Funktion nur kleiner.
2. Bei Polynomen können alle Exponenten von Subtrahenden **nicht** auf den Grad des Polynoms hochgestuft werden. Das würde die Funktion zwar kleiner machen, aber unter Umständen wird dadurch die Funktion negativ.
3. Bei Funktionen die kein Polynom sind, können andere Methoden zum Abschätzen vorteilhaft sein, z.B. das Benutzen von monoton-wachsenden Funktionen (Logarithmieren, Potenzieren, Wurzelziehen, etc.).

Wachstum von Funktionen

Bestimmen von Klassen

$$\frac{3n^2 - 5n\log n + 23n - 40}{3n\log n - 6n} \cdot \log n \in \Theta(n)$$

Beweis (O-Notation)

$$\frac{3n^2 - 5n\log n + 23n - 40}{3n\log n - 6n} \cdot \log n$$

Wachstum von Funktionen

Bestimmen von Klassen

$$\frac{3n^2 - 5n\log n + 23n\ - 40}{3n\log n - 6n} \cdot \log n \in \Theta(n)$$

Beweis (O-Notation)

$$\frac{3n^2 - 5n\log n + 23n\ - 40}{3n\log n - 6n} \cdot \log n \leq \frac{3n^2 + 23n}{3n\log n - 6n} \cdot \log n$$

Wachstum von Funktionen

Bestimmen von Klassen

$$\frac{3n^2 - 5n\log n + 23n\ - 40}{3n\log n - 6n} \cdot \log n \in \Theta(n)$$

Beweis (O-Notation)

$$\frac{3n^2 - 5n\log n + 23n\ - 40}{3n\log n - 6n} \cdot \log n \leq \frac{3n^2 + 23n}{3n\log n - 6n} \cdot \log n \leq \frac{26n^2}{3n\log n - 6n} \cdot \log n$$

Wachstum von Funktionen

Bestimmen von Klassen

$$\frac{3n^2 - 5n\log n + 23n\ - 40}{3n\log n - 6n} \cdot \log n \in \Theta(n)$$

Beweis (O-Notation)

$$\frac{3n^2 - 5n\log n + 23n\ - 40}{3n\log n - 6n} \cdot \log n \leq \frac{3n^2 + 23n}{3n\log n - 6n} \cdot \log n \leq \frac{26n^2}{3n\log n - 6n} \cdot \log n$$

$$\frac{26n^2}{3n\log n - 6n} \cdot \log n \leq \frac{26n^2}{3n\log n - n\log n} \cdot \log n$$

Ab $n_0 \geq 2^6 = 32$, da $6 \leq \log n$ gelten muss!

Wachstum von Funktionen

Bestimmen von Klassen

$$\frac{3n^2 - 5n\log n + 23n\ - 40}{3n\log n - 6n} \cdot \log n \in \Theta(n)$$

Also $n_0 \geq 32$ und $c_1 = 13$.

Beweis (O-Notation)

$$\frac{3n^2 - 5n\log n + 23n\ - 40}{3n\log n - 6n} \cdot \log n \leq \frac{3n^2 + 23n}{3n\log n - 6n} \cdot \log n \leq \frac{26n^2}{3n\log n - 6n} \cdot \log n$$

$$\frac{26n^2}{3n\log n - 6n} \cdot \log n \leq \frac{26n^2}{3n\log n - n\log n} \cdot \log n = 13n$$

Ab $n_0 \geq 2^6 = 32$, da $6 \leq \log n$ gelten muss!

Wachstum von Funktionen

Bestimmen von Klassen

$$\frac{3n^2 - 5n\log n + 23n - 40}{3n\log n - 6n} \cdot \log n \in \Theta(n)$$

Also $n_0 \geq 32$ und $c_1 = 13$.

Beweis (Ω-Notation)

$$\frac{3n^2 - 5n\log n + 23n - 40}{3n\log n - 6n} \cdot \log n$$

Wachstum von Funktionen

Bestimmen von Klassen

$$\frac{3n^2 - 5n\log n + 23n - 40}{3n\log n - 6n} \cdot \log n \in \Theta(n)$$

Also $n_0 \geq 32$ und $c_1 = 13$.

Beweis (Ω-Notation)

$$\frac{3n^2 - 5n\log n + 23n - 40}{3n\log n - 6n} \cdot \log n \geq \frac{3n^2 - 5n\log n + 23n - 40}{3n\log n} \cdot \log n$$

Wachstum von Funktionen

Bestimmen von Klassen

$$\frac{3n^2 - 5n\log n + 23n\ - 40}{3n\log n - 6n} \cdot \log n \in \Theta(n)$$

Also $n_0 \geq 32$ und $c_1 = 13$.

und $c_2 = \frac{2}{3}$.

Beweis (Ω-Notation)

$$\frac{3n^2 - 5n\log n + 23n\ - 40}{3n\log n - 6n} \cdot \log n \geq \frac{3n^2 - 5n\log n + 23n\ - 40}{3n\log n} \cdot \log n$$

$$\geq \frac{1}{3n}(3n^2 - 5n\log n + 23n\ - 40)$$

Ab $n_0 \geq 2$, da $40 \leq 20n$ gelten muss! →

$$\geq \frac{1}{3n}(3n^2 - 5n\log n + 23n\ - 20n) \geq \frac{1}{3n}(3n^2 - 5n\log n)$$

Ab $n_0 \geq 23$, da $5\log n \leq n$ gelten muss! →

$$\geq \frac{1}{3n}(3n^2 - n^2) \geq \frac{2}{3}n$$

Algorithmenentwurf

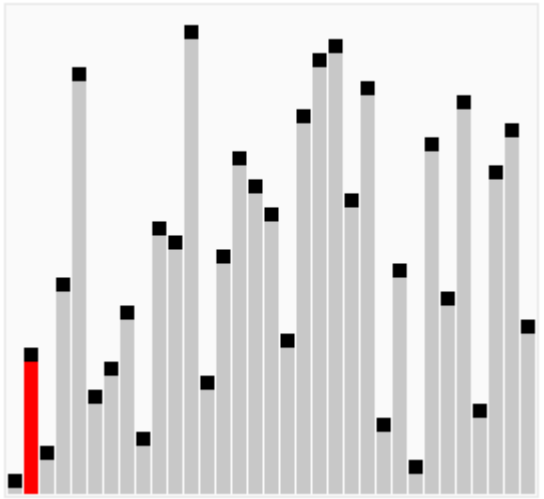
Sortieren mit zwei Stacks S_0, S_1 (und einer Objekt-Variablen)
Annahme: Stack S_1 enthält alle Elemente

1. Function COCKTAILSHAKERSORT(S_0, S_1)
2. For $j = 1$ to $\frac{n}{2}$ do
3. $item :=$ POP(S_0)
4. While !(IS_EMPTY(S_0))
5. if (TOP(S_0) $>$ $item$)
6. PUSH(S_1, POP(S_0)
7. else
8. PUSH(S_1, $item$)
9. $item :=$ POP(S_0)
10. Wiederhole 4. bis 9. und tausche dabei sowohl S_0 und S_1 als auch $>$ und $<$

Algorithmenentwurf

Sortieren mit zwei Stacks S_0, S_1 (und einer Objekt-Variablen)
Annahme: Stack S_1 enthält alle Elemente

1. Function COCKTAILSHAKERSORT(S_0, S_1)
2. For $j = 1$ to $\frac{n}{2}$ do
3. $item :=$ POP(S_0)
4. While !(IS_EMPTY(S_0))
5. if (TOP(S_0) $>$ $item$)
6. PUSH(S_1, POP(S_0))
7. else
8. PUSH(S_1, $item$)
9. $item :=$ POP(S_0)
10. Wiederhole 4. bis 9. und tausche dabei sowohl S_0 und S_1 als auch $>$ und $<$



Algorithmenentwurf – Laufzeit

Sortieren mit zwei Stacks S_0, S_1 (und einer Objekt-Variablen)
Annahme: Stack S_1 enthält alle Elemente

1. Function COCKTAILSHAKERSORT(S_0, S_1)
2. For $j = 1$ to $\frac{n}{2}$ do
3. $item :=$ POP(S_0)
4. While !(IS_EMPTY(S_0))
5. if (TOP(S_0) $>$ $item$)
6. PUSH(S_1, POP(S_0)
7. else
8. PUSH(S_1, $item$)
9. $item :=$ POP(S_0)
10. Wiederhole 4. bis 9. und tausche dabei sowohl S_0 und S_1 als auch $>$ und $<$.

Laufzeit:
- Schleife Zeile 2: $\sim n$ Iterationen
- Schleife Zeile 4: $\sim n$ Iterationen
- Zeile 10: Laufzeit von 4.-9.
- Zeilen 3, 5-9: O(1)

Insgesamt also:
$n \cdot \left(1 + 2 \cdot n \cdot O(1)\right) = O(n^2)$

Fragen?

Bei Fragen per Mails:
Immer an mich (aschmidt@ibr.cs.tu-bs.de) **und** Matthias (konitzny@ibr.cs.tu-bs.de).

Dabei beachten:
Fragen wie „Ich habe Thema X nicht verstanden. Kannst du das noch mal erklären?“ helfen weder euch, noch uns! – Warum?

Evaluation nicht vergessen!

(Läuft bis Mittag 16.02.2021)

Von Sándor Fekete via Aud <aud@ibr.cs.tu-bs.de>
Betreff **[Aud] Zwei Umfragen**
Antwort an Sándor Fekete <s.fekete@tu-bs.de>
An Sándor Fekete via Aud <aud@ibr.cs.tu-bs.de>
Kopie (CC) Sándor Fekete <s.fekete@tu-bs.de>

Liebe Teilnehmende,

wie schon in der Vorlesung vorhin angekündigt - hier die Links
und Infos zu zwei Umfragen:

(1) Die „normale“ elektronische Vorlesungsevalutation findet sich hier:

https://umfragen.tu-bs.de/evasys/online.php?pswd=LSF9Y

(Das ist die „übliche“ Auswertung, wie in jeder Veranstaltung und in jedem Jahr.)

HINWEIS: Die Nummern für Frage 10.16 („Wer hat die Übungsgruppe geleitet?“)
finden sich hier: https://aud.ibr.cs.tu-bs.de/index.php/organisation/

Viel Erfolg
und
frohes Schaffen!