Technische
Universität
Braunschweig

Algorithmen und Datenstrukturen – Übung #9

Fragestunde

Arne Schmidt
09.02.2023

Quiz!

Prüfung

Klausur
Algorithmen und Datenstrukturen
03.03.2023

Name: ....

Vorname: ....

Matr.-Nr.: ....

Studiengang: ....

☐ Bachelor ☐ Master ☐ Andere

Klausurcode:
Dieser wird benötigt, um das Ergebnis der Klausur abzurufen.

Hinweise:

- Bitte das Deckblatt in Druckschrift vollständig ausfüllen.
- Die Klausur besteht aus 12 Blättern, bitte auf Vollständigkeit überprüfen. Die Heftung darf nicht entfernt werden
- Erlaubte Hilfsmittel: keine
- Eigenes Papier ist nicht erlaubt.
- Die Rückseiten der Blätter dürfen beschrieben werden.
- Die Klausur ist mit 50 % der Punkte bestanden.
- Antworten, die *nicht* gewertet werden sollen, bitte deutlich durchstreichen. Kein Tippex verwenden!
- Mit *Bleistift* oder in *Rot* geschriebene Klausurteile können nicht gewertet werden.
- Werden mehrere Antworten gegeben, werten wir die mit der geringsten Punktzahl.
- Sämtliche Algorithmen, Datenstrukturen, Sätze und Begriffe beziehen sich, sofern nicht explizit anders angegeben, auf die in der Vorlesung vorgestellte Variante.
- Sofern nicht anders angegeben, sind alle Graphen als einfache Graphen zu verstehen.
- Die Bearbeitungszeit für die Klausur beträgt 120 Minuten.

Fragerunde

Themen

- Laufzeiten dynamische

  Datenstrukturen

- Laufzeitanalyse

- Wachstum von Funktionen
  - Vergleichen von Klassen
  - Bestimmen der Klassen
  - Beweise

- Quicksort

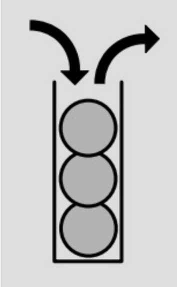
- Algorithmenverständnis

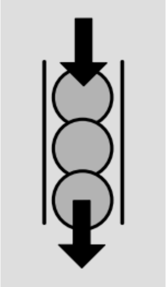
- (Induktionsbeweise)

Laufzeiten
dynamische Datenstrukturen

Laufzeiten – dynamische Datenstrukturen (ohne Sortierung)

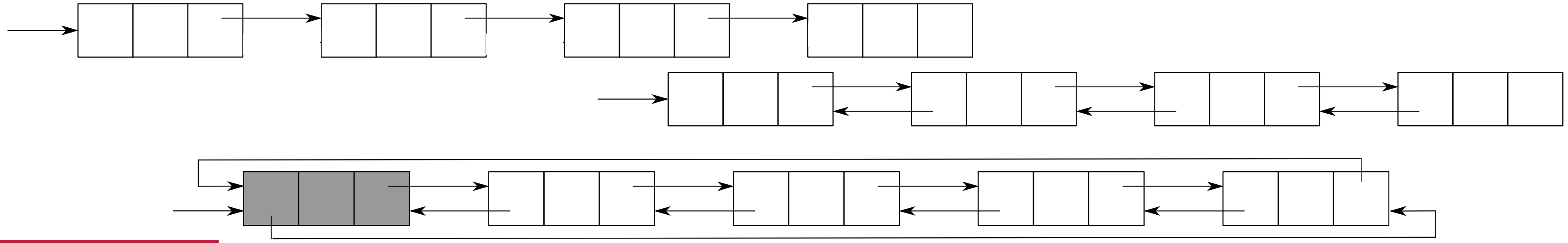
Datentyp	Stack	Queue
Einfügen	$O(1)$	$O(1)$
Nächstes Element	$O(1)$	$O(1)$
Löschen	$O(1)$	$O(1)$



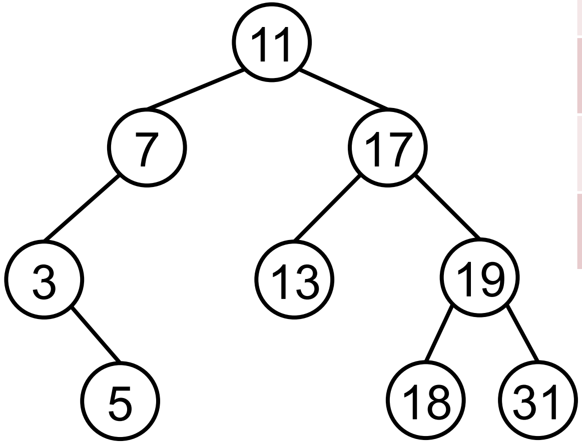


Laufzeiten – dynamische Datenstrukturen (ohne Sortierung)

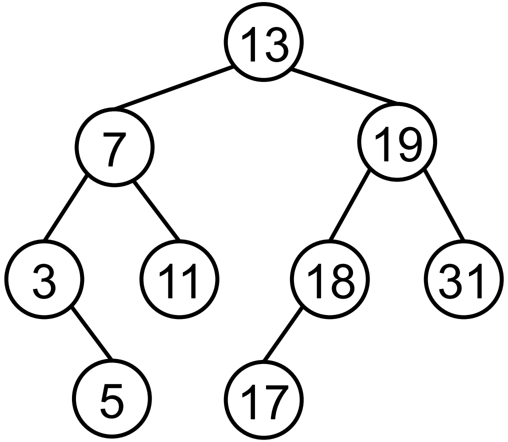
Datentyp	Listen		
Subtyp	Einfach	Doppelt	Zyklisch
Suchen	$O(n)$	$O(n)$	$O(n)$
Einfügen	$O(1)$	$O(1)$	$O(1)$
Löschen	$O(n)$	$O(1)$	$O(1)$
Traversierung	$O(n)$	$O(n)$	$O(n)$



Laufzeiten – dynamische Datenstrukturen (mit Sortierung)



Datentyp	Suchbäume	
Subtyp	-	AVL
Suchen	$O(h)$	$O(\log n)$
Einfügen	$O(h)$	$O(\log n)$
Löschen	$O(h)$	$O(\log n)$
Traversierung	$O(n)$	$O(n)$



Frage: Wie schnell kann ein bel. bin. Suchbaum in einen AVL-Baum umstrukturiert werden?

Laufzeiten – dynamische Datenstrukturen (partielle Sortierung)

Datentyp	(Max)Heaps	Fibonacci-Heaps
Einfügen	$O(\log n)$	$O(1)$
Löschen	$O(\log n)$	$O(\log n)$*
Minimum/Maximum	$O(1)$	$O(1)$
Extrahiere Min/Max	$O(\log n)$	$O(\log n)$*

*: Amortisierte Laufzeit, d.h. Durchschnitt über viele solche Operationen

Laufzeitanalyse

Sortieren mit Bubblesort

Algorithm 1: Bubblesort(A, n)

begin
  for $j := n - 1$ *DOWNTO* 1 **do**
    for $i := 1$ *TO* j **do**
      if $A[i] < A[i+1]$ **then**
        vertausche $(A[i], A[i+1])$

Klausur WiSe 11/12

Laufzeit:

$\#Iterationen * \#Iterationen * Box$

$\#Iterationen * \#Iterationen * O(1)$

$\#Iterationen * O(n) * O(1)$

$O(n) * O(n) * O(1)$

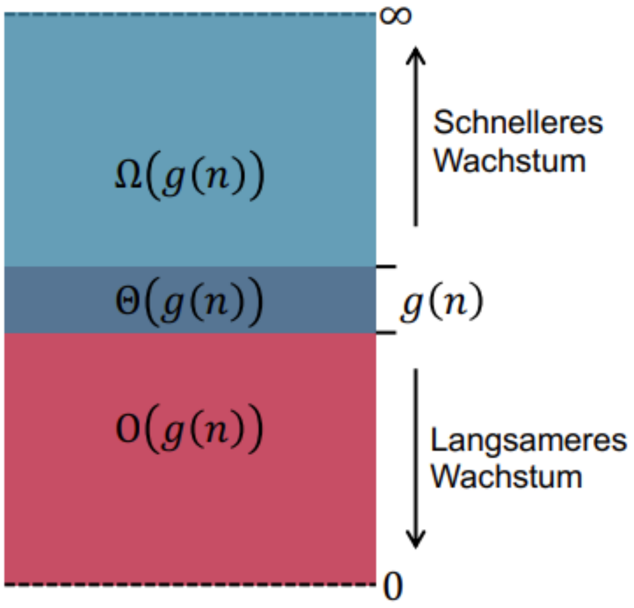
$\Rightarrow O(n^2)$

Wachstum von Funktionen

Wachstum von Funktionen (Relationen von Klassen)

Wachstum von Funktionen

Vergleichen von Klassen



Hierarchie-Ausschnitt:

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

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 der Klasse)

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

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

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

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)

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

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

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

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

Ab $n_0 \geq 23$, da $5\log \mathrm{n} \leq n$ gelten muss!

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

Wachstum von Funktionen (Beweise)

Satz 3.12

Seien $f, g\colon \mathbb{N} \to \mathbb{R}$, dann gilt $f(n) \in O\big(g(n)\big) \Leftrightarrow g(n) \in \Omega\big(f(n)\big)$

$f(n) \in O\big(g(n)\big)$

$\Leftrightarrow \exists c \in \mathbb{R}^+, n_0 \in \mathbb{N}\colon\ \forall n \geq n_0\colon \quad 0 \leq f(n) \leq c \cdot g(n)$

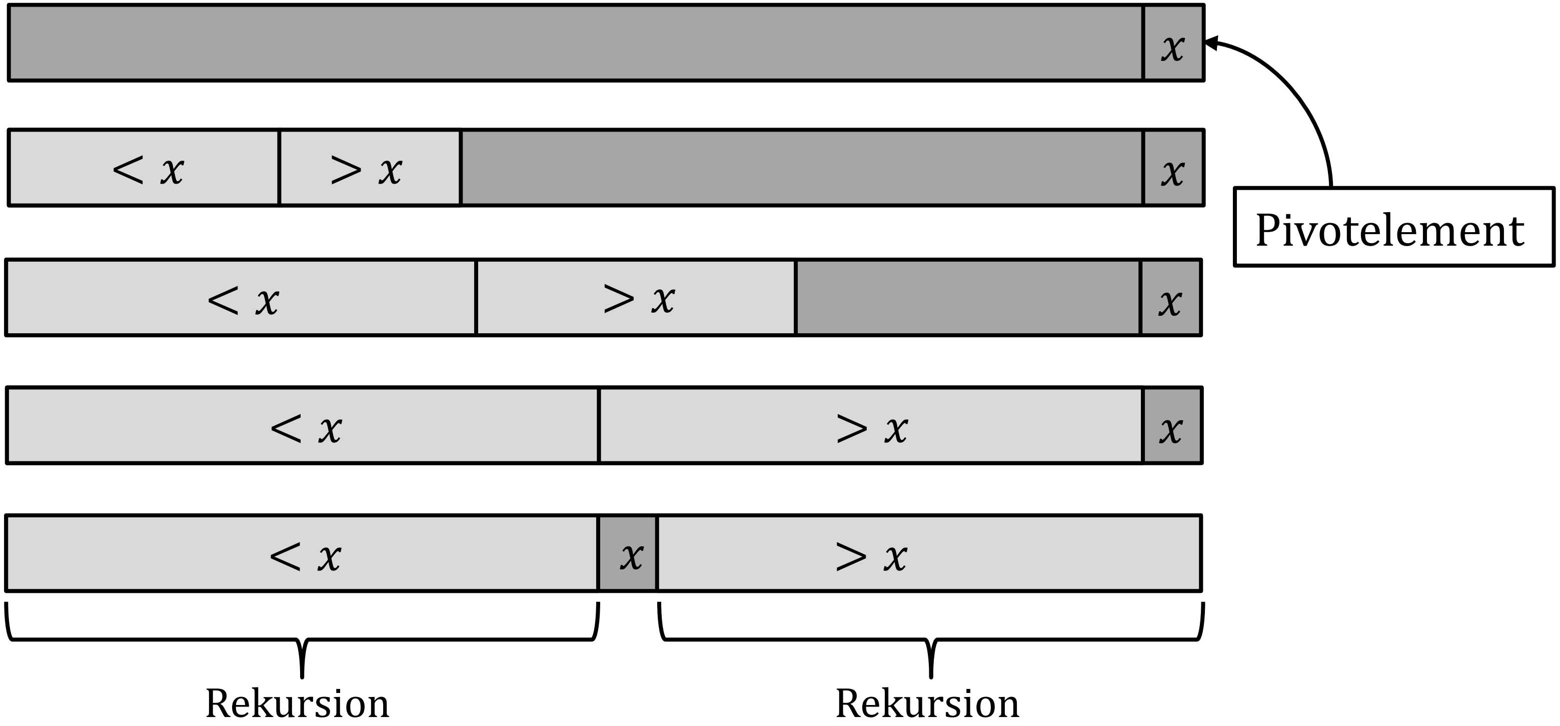
$\Leftrightarrow \exists c \in \mathbb{R}^+, n_0 \in \mathbb{N}\colon\ \forall n \geq n_0\colon \quad 0 \leq \dfrac{1}{c} \cdot f(n) \leq g(n)$

$\Leftrightarrow \exists c' \in \mathbb{R}^+, n_0 \in \mathbb{N}\colon\ \forall n \geq n_0\colon \quad 0 \leq c' \cdot f(n) \leq g(n)$ (nämlich $c' = \frac{1}{c}$)

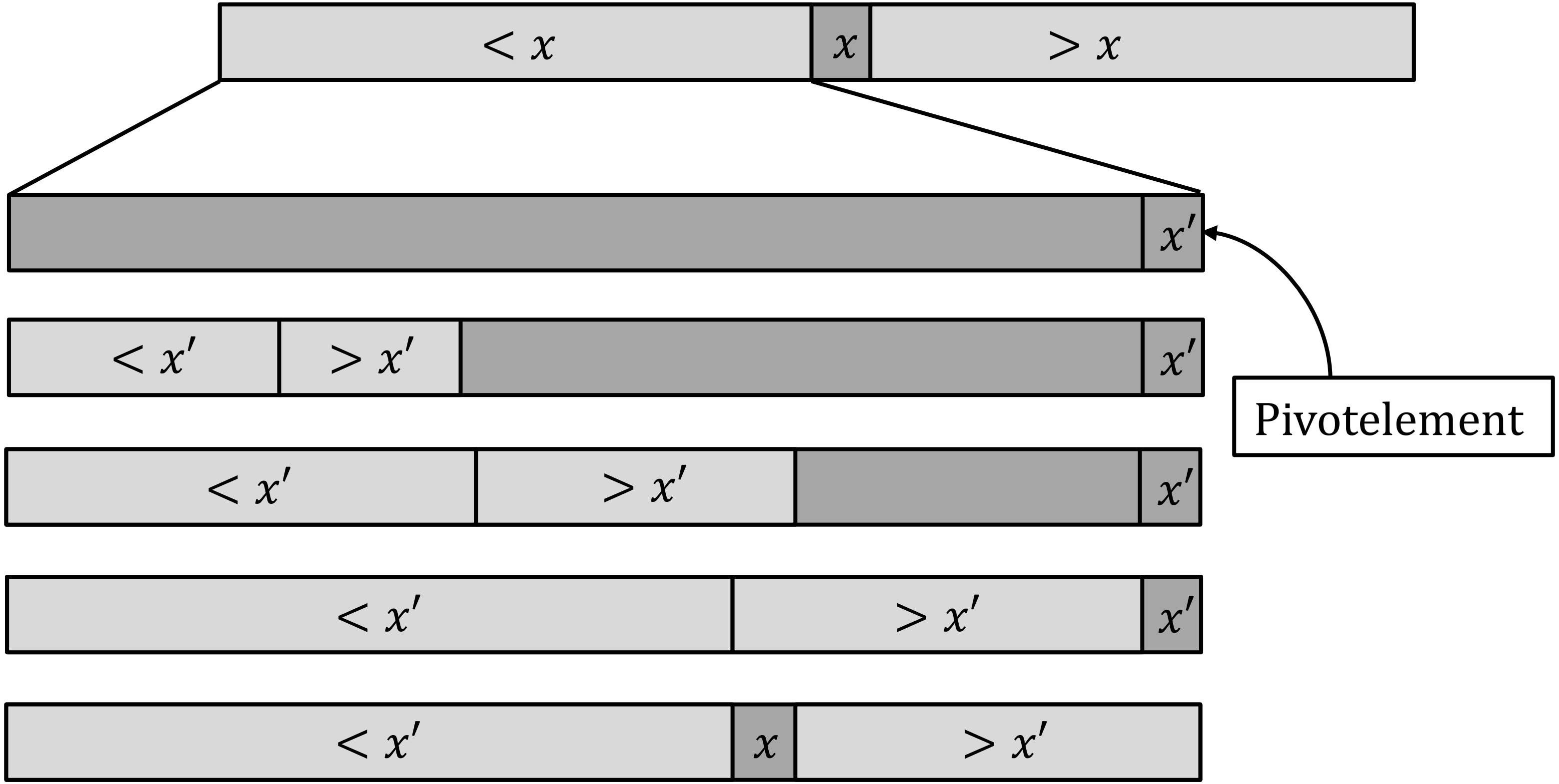
$\Leftrightarrow g(n) \in \Omega\big(f(n)\big)$

Quicksort

Quicksort – Prinzip



Quicksort – Prinzip



Algorithmenverständnis

Algorithmenverständnis

Gegeben zwei Stacks S_0, S_1 (und eine Objekt-Variablen)
Annahme: Stack S_1 ist leer.

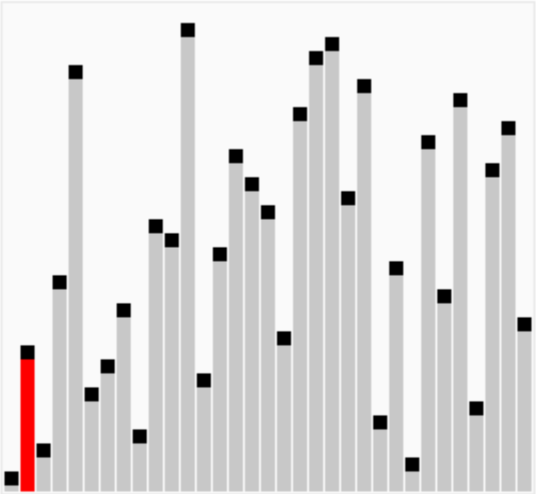
Was tut dieser Algorithmus?

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

Algorithmenverständnis

Gegeben zwei Stacks S_0, S_1 (und eine Objekt-Variablen)
Annahme: Stack S_1 ist leer.

1. Function COCKTAILSHAKER(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 $<$



Algorithmenverständnis – Laufzeit

Gegeben zwei Stacks S_0, S_1 (und eine Objekt-Variablen)
Annahme: Stack S_1 ist leer.

1. Function COCKTAILSHAKER(S_0, S_1)
2. For $j = 1$ to $\frac{n}{2}$ do
3. $\quad item :=$ POP(S_0)
4. $\quad$ While !(IS_EMPTY(S_0))
5. $\quad\quad$ if (TOP(S_0) $>$ $item$)
6. $\quad\quad\quad$ PUSH(S_1, POP(S_0))
7. $\quad\quad$ else
8. $\quad\quad\quad$ PUSH(S_1, $item$)
9. $\quad\quad\quad$ $item :=$ POP(S_0)
10. $\quad$ 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:

$$n \cdot \big(1 + 2 \cdot n \cdot O(1)\big) = O(n^2)$$

Induktionsbeweise

Handshake-Lemma

Satz. Sei $G = (V, E)$ ein Graph. Dann besitzt G eine gerade Anzahl an ungeraden Knoten.

Beweis per Induktion über die Anzahl an Kanten m.

IA: Graph ohne Kanten ($m = 0$) besitzt nur gerade Knoten (alle Grad 0).

IV: Annahme gelte für beliebiges, aber festes $m \in \mathbb{N}$.

IS:

- Betrachte Graph mit $m + 1$ Knoten.
- Entferne eine Kante $e = \{u, v\}$.
- Nach IV besitzt $G \setminus e$ gerade Anzahl ungerader Knoten.
- Nach Einfügen von e gilt Eigenschaft wieder (siehe Tabelle).

Ohne e		Mit e		Änderung
u	v	u	v	-
0	0	1	1	+2
1	0	0	1	+0
1	1	0	0	-2

Grad modulo 2

Fragen?

Bei Fragen per Mails:
Immer an mich (aschmidt@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?

Sprechstunde

Eintragen:

- Maximal in eine Gruppe
- Vor- und Nachname (oder falls Datenschutzbedenken vorliegen: 1. Buchstabe Vor- und Nachname gefolgt von Gruppennummer und letzte Matrikelziffer)

Sprechstunde der Tutoren AuD Winter 22/23

	Wann	Do., 23.02.2023	Fr., 24.02.2023	Mo., 27.02.2023	Di., 28.02.2023	Mi., 01.03.2023
		13:00 Uhr	13:00 Uhr	11:30 Uhr	13:00 Uhr	13:00 Uhr
	Tutor	Kevin Meier	Till Bohmfalk	Patrick Blumenberg	Tilo Hoitz	Dennis Dinh
	Mail	kmeier@ibr.cs.tu-bs.de	bohmfalk@ibr.cs.tu-bs.de	blumenbe@ibr.cs.tu-bs.de	hoitz@ibr.cs.tu-bs.de	dinh@ibr.cs.tu-bs.de
	Belegung	3,57 %	3,57 %	0,00 %	0,00 %	0,00 %
Teilnehmende	1	Arne Schmidt	AS13			
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	10					

Link zur Anmeldung per Mailingliste!

Im Sommer…

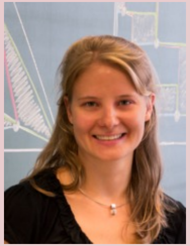
…gibt es wieder zwei Veranstaltungen für den Bachelor-Bereich:

Algorithmen und Datenstrukturen 2

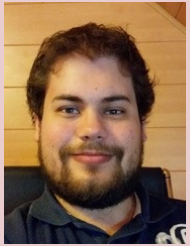
Prof. Sándor Fekete

- Einführung in Komplexitätstheorie
- Heuristiken
- Exakte Methoden
- Approximationen
- Hashing

Netzwerkalgorithmen



Linda Kleist



Arne Schmidt

- Kostenminimale aufspannende Bäume
- Kürzeste Wege
- Maximale Flüsse
- Kardinalitätsmaximales Matching

Viel Erfolg
und
frohes Schaffen!