FernUniversität in Hagen

# Lösungsvorschläge zur Klausur 1661 „Datenstrukturen I“

**17.02.2018**

# Aufgabe 1

(a)

Die Funktionen

$$f(n) = \begin{cases} -n \text{ für gerade } n \\ \\ n \text{ für ungerade } n \end{cases} \quad \text{und} \quad g(n) = f(n+1)$$

erfüllen z.B. diese Bedingung, denn

$$\begin{aligned} \forall n_0 \in \mathbb{N}\ \exists k, m \in \mathbb{N}: \quad & 2k = m > n_0 \wedge f(m) = -m \wedge g(m) = m+1 \\ & \wedge f(m+1) = m+1 \wedge g(m+1) = -(m+2) \end{aligned}$$

Also gilt weder $f(n) = O(g(n))$ noch $g(n) = O(f(n))$.

(b)

Wegen $f(n) = o(g(n))$ gilt $f(n) = O(g(n))$, also $f(n) + g(n) \leq 2g(n) = O(g(n))$ und $S(n) = O(g(n))$. Da $g(n) \leq f(n) + g(n) = 2 \cdot S(n)$, ist auch $g(n) = O(S(n))$, also $S(n) = \Theta(g(n))$. $S(n)$ wächst damit gleich schnell wie $g(n)$.

Es reicht hier nicht, nur zu sagen, dass $S(n) = O(g(n))$ ist.

(c)

Da $g(n)$ schneller wächst als $f(n)$, ist $\frac{f(n)}{g(n)}$ eine Nullfolge. Daher ist auch $\sqrt{\frac{f(n)}{g(n)}}$ Nullfolge, und es gilt $\frac{f(n)}{T(n)} = \sqrt{\frac{f(n)}{g(n)}} = \frac{T(n)}{g(n)}$. Also wächst $T(n)$ schneller als $f(n)$ und langsamer als $g(n)$.

(d)

$O(n^2)$ enthält alle Funktionen, die höchstens so schnell wachsen wie $f(n) = n^2$. Die Aufgabenstellung besagt nun, dass die Laufzeit von $A$ in $O(n^2)$ liegt oder schneller wächst als $f(n) = n^2$. Also erfüllen alle möglichen Laufzeitfunktionen die Aussage.

(e)

Diese Gleichung ist falsch, denn aus ihr kann eine falsche Aussage hergeleitet werden:

$$\exists n_0 \in \mathbb{N}, c \in \mathbb{R}^+\ \forall n > n_0: \quad n^3 2^n + 6n^2 3^n \leq c\, n^3 2^n$$

$$\Leftrightarrow \quad \exists n_0 \in \mathbb{N}, c \in \mathbb{R}^+\ \forall n > n_0: \quad 6n^2 3^n \leq (c-1) n^3 2^n$$

$$\Leftrightarrow \quad \exists n_0 \in \mathbb{N}, c \in \mathbb{R}^+\ \forall n > n_0: \quad 3^n \leq \tfrac{c-1}{6} n 2^n$$

$$\Leftrightarrow \quad \left(\tfrac{3}{2}\right)^n = O(n).$$

Die zuletzt hergeleitete Behauptung ist jedoch offensichtlich falsch.

(f)

Entsprechend der Definition der O-Notation prüfen wir zur Beantwortung der Frage

$$f(n) \stackrel{?}{=} O(g(n))$$

jeweils, ob Konstanten $c$ und $n_0$ existieren, für die gilt:

$$\forall n \geq n_0 : f(n) \leq c \cdot g(n)$$

$$(n+1)! \leq c \cdot n! \Leftrightarrow$$

$$(n+1) \cdot n! \leq c \cdot n! \Leftrightarrow$$

$$c \geq n+1$$

Der Wert von $c$ muss also größer sein als $n$, um die Ungleichung zu erfüllen. Deshalb ist es nicht möglich, eine entsprechende Konstante $c$ anzugeben: $(n+1)! \notin O(n!)$.

## Aufgabe 2 Hashing

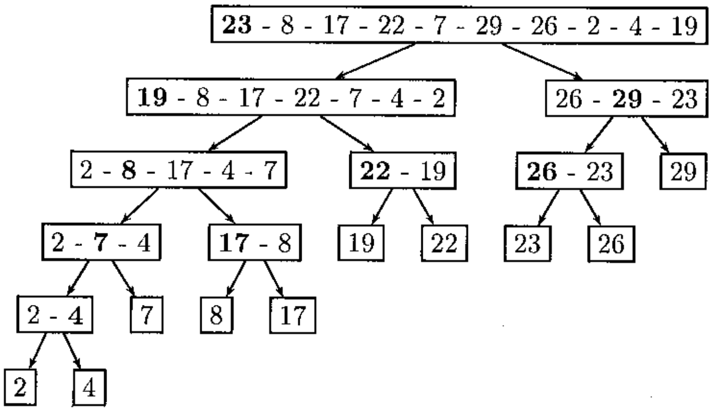
| Wert | $k$ | $k^2$ | $h_0$ | $h_1$ | $h_2$ |
|---|---|---|---|---|---|
| Michael | 25 | 625 | 2 | | |
| Daniel | 19 | 361 | 6 | | |
| Ronny | 47 | 2209 | 0 | | |
| Franziska | 25 | 625 | 2 | 3 | |
| Theresa | 33 | 1089 | 8 | | |
| Birgit | 29 | 841 | 4 | | |
| Sebastian | 26 | 676 | 7 | | |
| Christian | 29 | 841 | 4 | 5 | |
| Kathrin | 32 | 1024 | 2 | 3 | 1 |

Es ergibt sich die folgende Hashtabelle:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Ronny | Kathrin | Michael | Franziska | Birgit | Christian | Daniel | Sebastian | Theresa | |

# Aufgabe 3 Sortieren

(a)

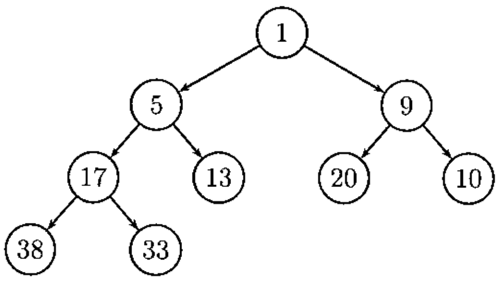



(b)

Heapsort wird gestartet

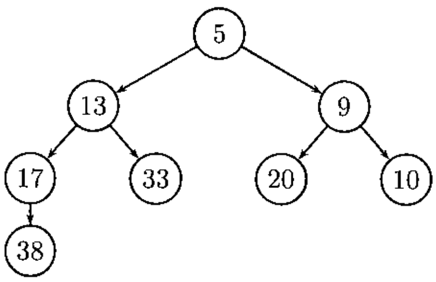
Arrayeinbettung : | 13 | 17 | 10 | 33 | 5 | 20 | 9 | 38 | 1 |

Initialer Heap erzeugt




Arrayeinbettung : | 1 | 5 | 9 | 17 | 13 | 20 | 10 | 38 | 33 |

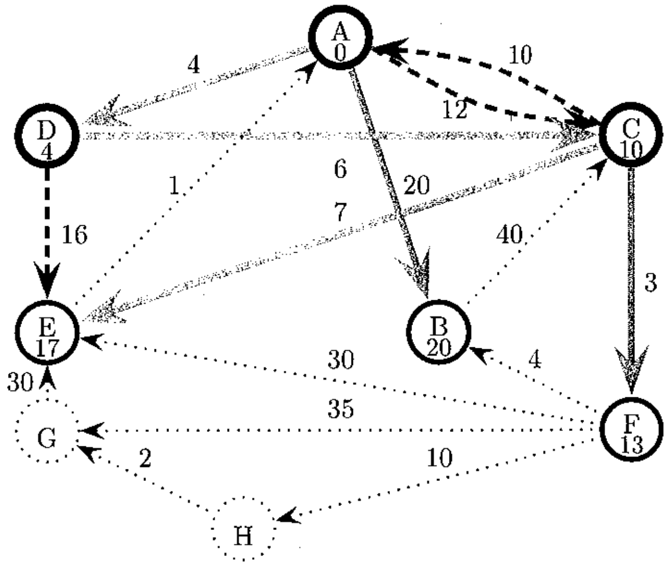
Element 1 verarbeitet




Sortierte Folge: 1

Arrayeinbettung : | 5 | 13 | 9 | 17 | 33 | 20 | 10 | 38 || 1 |

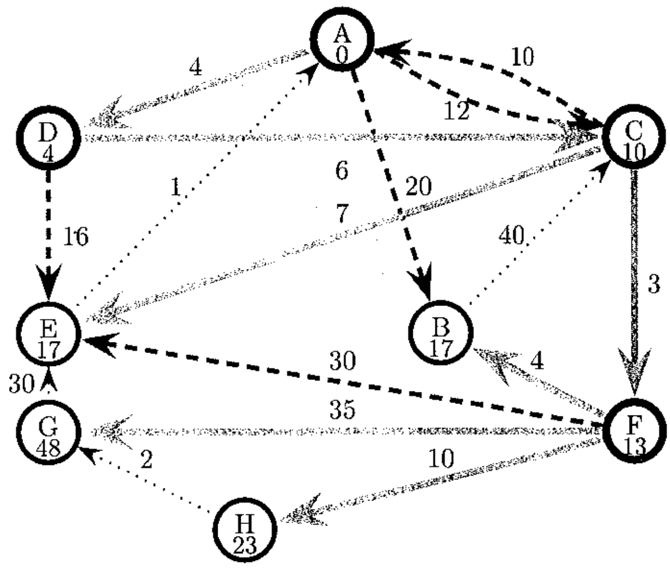
Nach Verarbeitung des Knotens C




Aktionen beim Verarbeiten des Knotens F

- Färbe Knoten F grün
- Kürzerer Weg zu Knoten B gefunden, neue Distanz ist:17
- Färbe Kante (A,B) gelb
- Färbe Kante (F,B) rot
- Färbe Kante (F,E) gelb
- Färbe Knoten G gelb, setze Distanz auf 48
- Färbe Kante (F,G) rot
- Färbe Knoten H gelb, setze Distanz auf 23
- Färbe Kante (F,H) rot

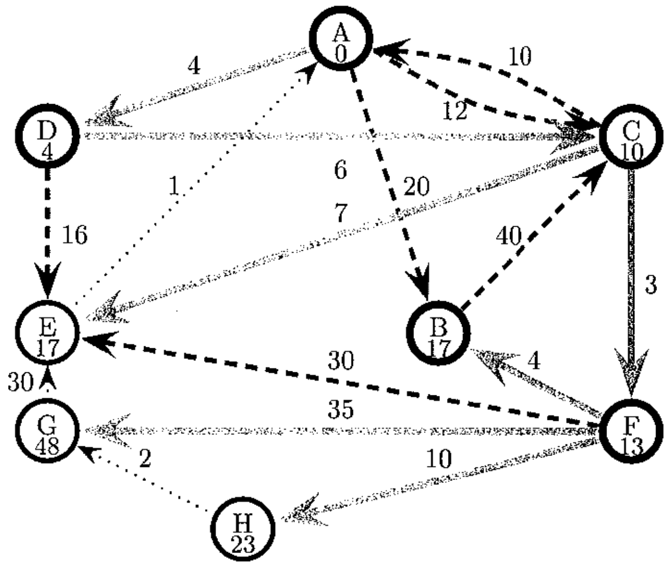
Nach Verarbeitung des Knotens F



Aktionen beim Verarbeiten des Knotens B

- Färbe Knoten B grün
- Färbe Kante (B,C) gelb

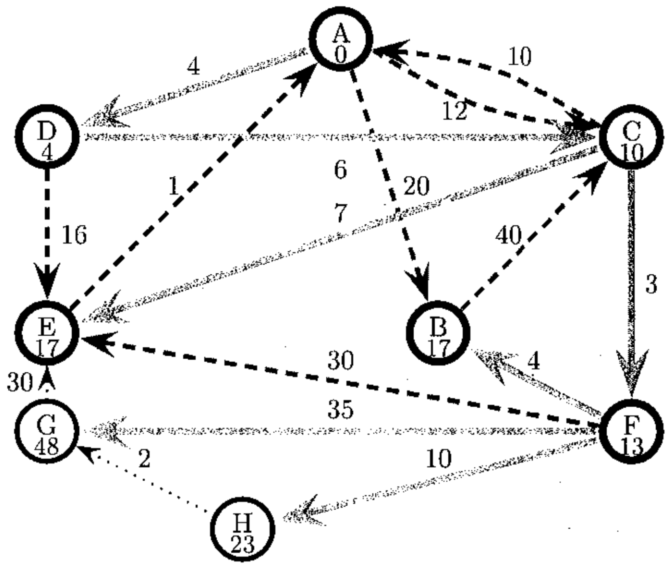
Nach Verarbeitung des Knotens B




Aktionen beim Verarbeiten des Knotens E

- Färbe Knoten E grün
- Färbe Kante (E,A) gelb

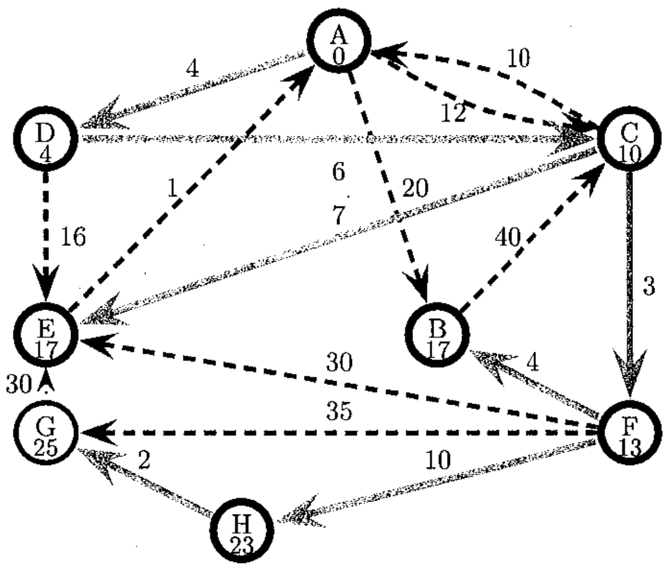
Nach Verarbeitung des Knotens E



Aktionen beim Verarbeiten des Knotens H

- Färbe Knoten H grün
- Kürzerer Weg zu Knoten G gefunden, neue Distanz ist:25
- Färbe Kante (F,G) gelb
- Färbe Kante (H,G) rot

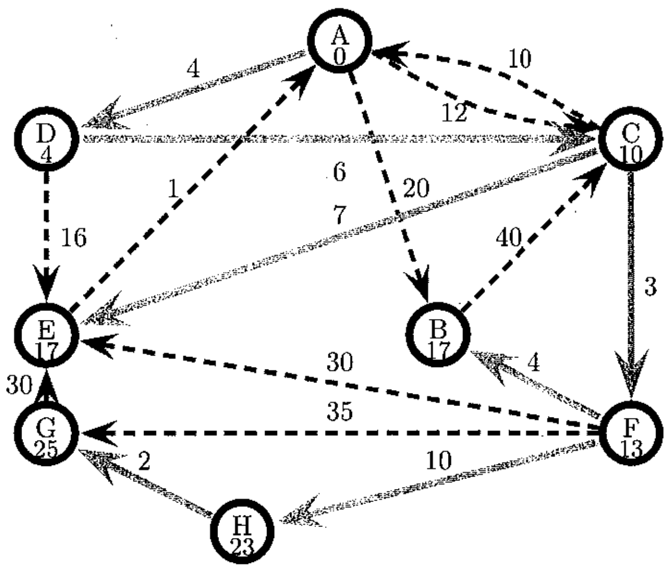
Nach Verarbeitung des Knotens H




Aktionen beim Verarbeiten des Knotens G

- Färbe Knoten G grün
- Färbe Kante (G,E) gelb

Nach Verarbeitung des letzten Knotens ergibt sich folgendes Bild:



Eine tabellarische Lösung kann wie folgt aussehen:

| Schritt | grün | gelbe Knoten | rote Kanten | gelbe Kanten | Updates |
|---|---|---|---|---|---|
| 1 | A | B(20),C(12),D(4) | (A,B),(A,C),(A,D) | | |
| 2 | D | E(20) | (D,C),(D,E) | (A,C) | C(10) |
| 3 | C | F(13) | (C,E),(C,F) | (C,A),(D,E) | E(17) |
| 4 | F | G(48),H(23) | (F,B),(F,G),(F,H) | (A,B),(F,E) | B(17) |
| 5 | B | | | (B,C) | |
| 6 | E | | | (E,A) | |
| 7 | H | | (H,G) | (F,G) | G(25) |
| 8 | G | | | (G,E) | |

## Aufgabe 5 Deckblatt

Hier erhalten Sie den Punkt, wenn Sie beide Klausurdeckblätter korrekt und vollständig ausgefüllt haben, also Namen, Matrikelnummer und Adresse korrekt eingetragen und genau diejenigen Aufgaben markiert haben, die Sie auch bearbeitet haben.