

Algorithmen 1 SS 2013 – Tutorium 7

7. Tutorium

Sarah Lutteropp

4. Juni 2013

Übersicht

1 6. Übungsblatt

2 Quickselect

3 Heaps

4 Kreativaufgabe

6. Übungsblatt

Eure Tutorin ...

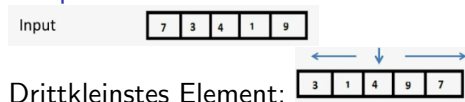
Hatte noch keine Zeit für das Blatt und macht das in dem Vorlesungsblock vor dem Tutorium.

Quickselect

Was tut das?

- Finde k -kleinstes Element, tue dieses an Position k
- Partitioniere um dieses Element herum

Beispiel



Grundgedanke dahinter

- Rate Element aus dem Array
- Benutze Partition aus Quicksort mit dem geratenen Pivot
- Falls nicht fertig: Mache binäre Suche, um das passende Element zu finden (entweder im linken oder im rechten Teil)

Quickselect

\approx quicksort mit einseitiger Rekursion

Function $\text{select}(s : \text{Sequence of Element}; k : \mathbb{N}) : \text{Element}$

assert $|s| \geq k$

pick $p \in s$ uniformly at random // pivot key

$a := \langle e \in s : e < p \rangle$

if $|a| \geq k$ **then return** $\text{select}(a, k)$ //

a

 k

$b := \langle e \in s : e = p \rangle$

if $|a| + |b| \geq k$ **then return** p //

a	$b = \langle p, \dots, p \rangle$
-----	-----------------------------------

 k

$c := \langle e \in s : e > p \rangle$

return $\text{select}(c, k - |a| - |b|)$ //

a	b	c
-----	-----	-----

 k

Laufzeit: $\mathcal{O}(|s|)$

Binäre Heaps

Binärer Heap

Binäre Heaps werden hauptsächlich als Prioritätswarteschlange verwendet.

- Teilsortiert (Max-Heap/Min-Heap)
- Immer von links auffüllen
- Keine Lücken

Aufgabe

Höhe

Welche Höhe hat ein binärer Heap mit n Elementen?

Aufgabe

Höhe

Welche Höhe hat ein binärer Heap mit n Elementen?

$$h = \lfloor \log(n) \rfloor$$

Aufgabe

Höhe

Welche Höhe hat ein binärer Heap mit n Elementen?

$$h = \lfloor \log(n) \rfloor$$

Elemente

Wieviel Elemente hat ein binärer Heap mindestens und höchstens, wenn er die Höhe n hat?

Aufgabe

Höhe

Welche Höhe hat ein binärer Heap mit n Elementen?

$$h = \lfloor \log(n) \rfloor$$

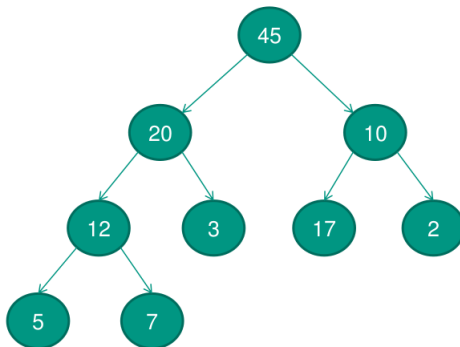
Elemente

Wieviel Elemente hat ein binärer Heap mindestens und höchstens, wenn er die Höhe n hat?

$$\text{mindestens: } 2^n, \text{ höchstens: } 2^{n+1} - 1$$

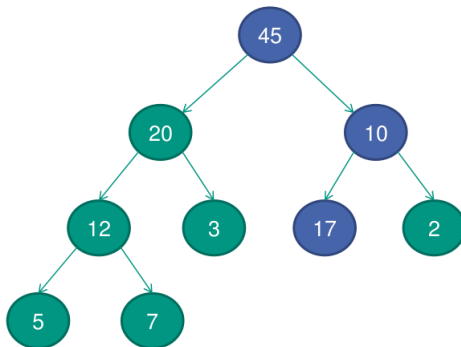
Aufgabe

Ist das ein binärer Heap?



Aufgabe

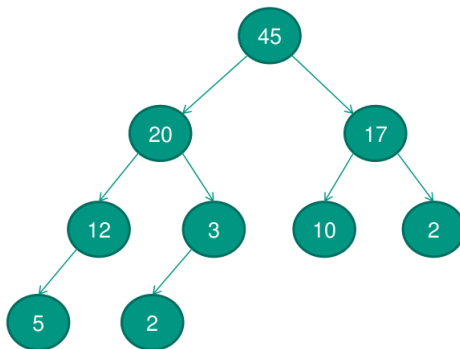
Ist das ein binärer Heap?



Nein! Knoteninhalte sind in falscher Ordnung!

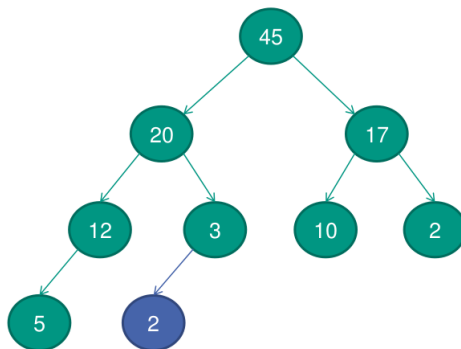
Aufgabe

Ist das ein binärer Heap?



Aufgabe

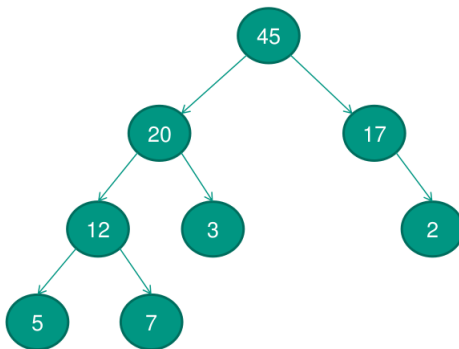
Ist das ein binärer Heap?



Nein! Letzte Schicht nicht von links aufgefüllt!

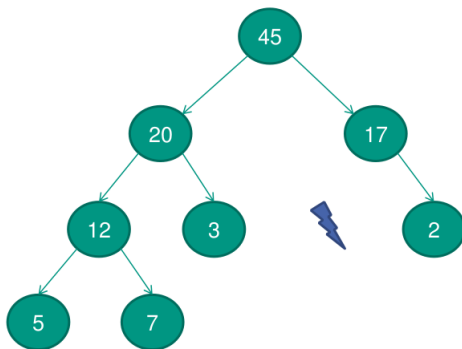
Aufgabe

Ist das ein binärer Heap?



Aufgabe

Ist das ein binärer Heap?



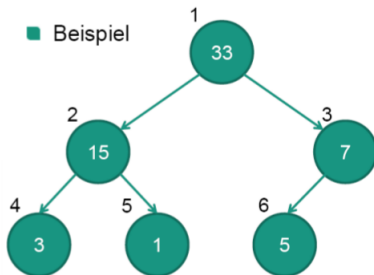
Nein! Schicht unvollständig!

Binäre Heaps – Darstellung

- Darstellung als **Array**

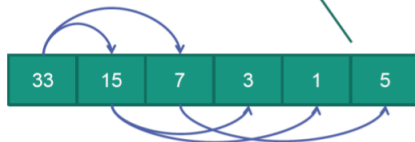
- Kinder des Elements $A[i]$ sind $A[2i]$ und $A[2i + 1]$

- Beispiel



Max-Heap
mit $A.\text{heapgröße} = 6$

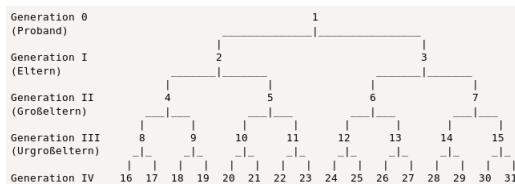
Keine Lücken im Array!
(bedingt durch
Baumstruktur und -konstruktion)



Was ist der Index des Elternknotens?

Exkurs

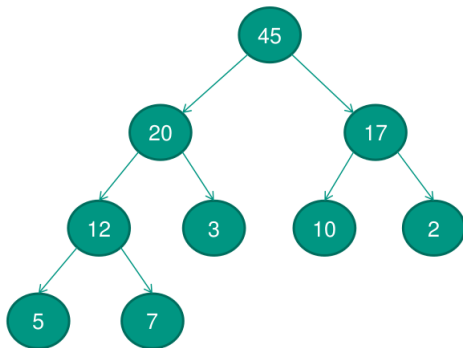
Kekule-Nummerierung



Unterschied zu Heaps: Eltern und Kinder-Bezeichnung vertauscht

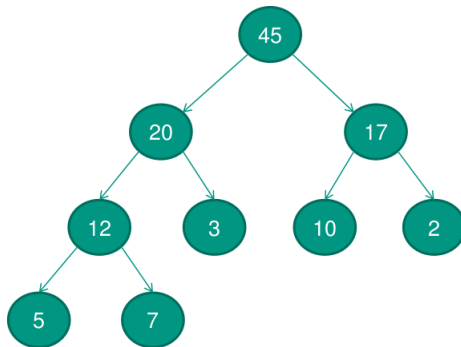
Aufgabe

Bestimme das dazugehörige Array



Lösung

Bestimme das dazugehörige Array



45	20	17	12	3	10	2	5	7
----	----	----	----	---	----	---	---	---

Aufgabe

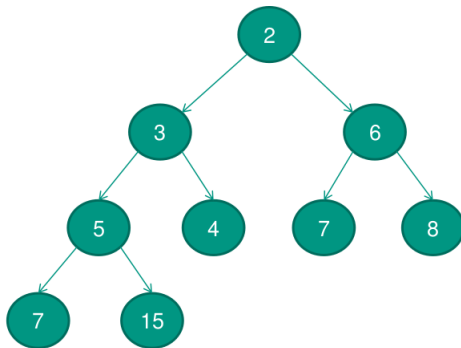
Bestimme den dazugehörigen binären Min-Heap



2	3	6	5	4	7	8	7	15
---	---	---	---	---	---	---	---	----

Lösung

Bestimme den dazugehörigen binären Min-Heap

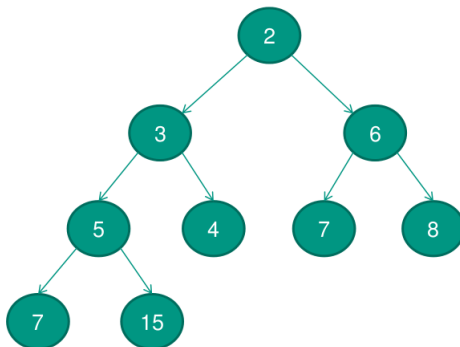


Binäre Heaps – ExtractMax/Min

- Letztes Element an erste Stelle schreiben
- Min/Max der Kinder bestimmen
- Ggf. tauschen, falls Heap-Eigenschaften verletzt
- Rekursiv mit dem Max/Min fortfahren

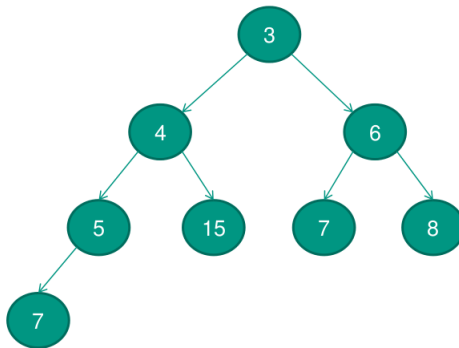
Aufgabe

Führe die Operation ExtractMin aus



Lösung

Führe die Operation ExtractMin aus

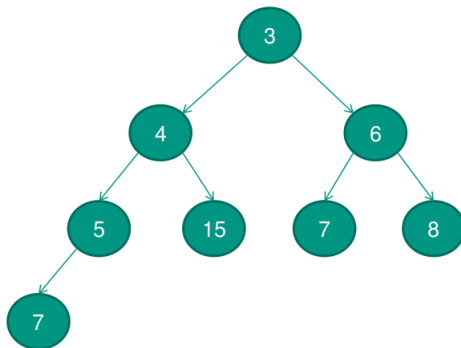


Binäre Heaps – Insert

- Element an letzter Stelle einfügen
- Vergleiche mit Elternknoten
- Tausche, falls Heap-Eigenschaften verletzt
- Wiederhole bis zur Wurzel

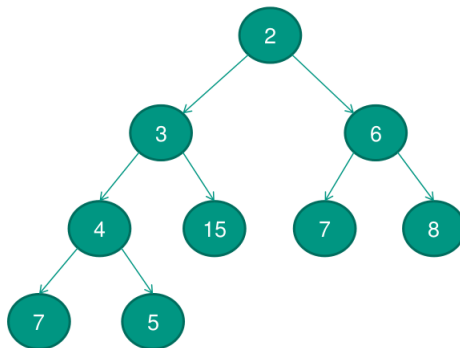
Aufgabe

Füge den Schlüssel 2 ein



Lösung

Füge den Schlüssel 2 ein



Binäre Heaps – Build in $\mathcal{O}(n)$

- Das ganze Array als (kaputten) Heap interpretieren
- Von den Blättern ausgehend die Heap-Eigenschaft wiederherstellen

Binäre Heaps – Sortieren

- Heap erstellen
- Wiederholen bis Heapgröße=1 (ExtractMax):
 - Maximum mit letztem Element aus unsortiertem Feld tauschen
 - Heap verkleinern
 - Heapeigenschaften wiederherstellen

Aufgabe

Führe auf folgendem Array Heapsort aus. Benutze dafür einen MaxHeap.

32	3	15	8	78	7
----	---	----	---	----	---

Lösung

- 1

78	32	15	3	8	7
----	----	----	---	---	---

 Heap-Eigenschaft hergestellt
- 2

7	32	15	3	8	78
---	----	----	---	---	----

 Maximum ans Ende
- 3

32	8	15	3	7	78
----	---	----	---	---	----

 Heap-Eigenschaft hergestellt
- 4

7	8	15	3	32	78
---	---	----	---	----	----

 Maximum ans Ende
- 5

15	8	7	3	32	78
----	---	---	---	----	----

 Heap-Eigenschaft hergestellt
- 6

3	8	7	15	32	78
---	---	---	----	----	----

 Maximum ans Ende
- 7

8	3	7	15	32	78
---	---	---	----	----	----

 Heap-Eigenschaft hergestellt
- 8

7	3	8	15	32	78
---	---	---	----	----	----

 Maximum ans Ende
- 9

3	7	8	15	32	78
---	---	---	----	----	----

 Maximum ans Ende

Kreativaufgabe

Pancake-Sorting

Gegeben sind n Pancakes in unterschiedlicher Größe und gestapelt. Man hat einen Pancake-Flipper zur Verfügung, mit dem man die obersten Pancakes umdrehen kann (bel.). Entwickelt einen schnellen Algorithmus um die Pancakes zu sortieren.

<http://www.youtube.com/watch?v=kk-DDgoXfk>

Bis zum nächsten Mal.



Anmerkung:

Die Heap-Folien sind größtenteils geklaut von: Benjamin Brandmüller - Algorithmen I Tutorium, 16.05.2011

Die Erklärung von Quickselect stammt aus: <http://www.youtube.com/watch?v=a0hyCdxGJvY>