Algorithmen I

Tutorium 1 - 5. Sitzung

Dennis Felsing

dennis.felsing@student.kit.edu
www.stud.uni-karlsruhe.de/~ubcqr/algo

2011-05-16

Heaps

Binäre Heaps

Definition

Ein **Binärer Heap** ist ein Array, dass sich als Binärbaum darstellen lässt. Bei **Min-Heaps** gilt als **Heap-Eigenschaft** $A[\textsc{Parent}(i)] \leq A[i]$, bei **Max-Heaps** gilt $A[\textsc{Parent}(i)] \geq A[i]$.

Wir arbeiten im folgenden mit Max-Heaps.

Wichtige Operationen

- $\textsc{Max-Heapify}$: Stelle Max-Heap-Eigenschaft her ($O(\log n)$)
- $\textsc{Build-Max-Heap}$: Erzeuge Max-Heap aus Array ($O(n)$)
- $\textsc{Heapsort}$: Sortiere ein Array (inplace, $O(n \log n)$)

Binäre Heaps

Aufgabe

Welcher Zusammenhang besteht zwischen sortierten Arrays und Heaps?

Lösung

- Aufsteigend sortiertes Array $\Rightarrow$ Min-Heap
- Absteigend sortiertes Array $\Rightarrow$ Max-Heap
- Umkehrung gilt nicht!

Erhalten der Heap-Eigenschaft

Situation

Die Binärbäume mit Wurzel $\textsc{Left}(i)$ und $\textsc{Right}(i)$ sind Max-Heaps. $A[i]$ kann aber kleiner als seine Kinder sein.

$\textsc{Max-Heapify}(A, i)$

```
 1  l = Left(i)
 2  r = Right(i)
 3  max = i
 4  if l ≤ A.heap-size und A[l] > A[max]
 5        max = l
 6  if r ≤ A.heap-size und A[r] > A[max]
 7        max = r
 8  if max ≠ i
 9        vertausche A[i] mit A[max]
10        Max-Heapify(A, max)
```

Erzeugen eines Heaps

BUILD-MAX-HEAP(A)

1 $A.heap\text{-}size = A.length$
2 **for** $i = \lfloor A.length/2 \rfloor$ **downto** 1
3 MAX-HEAPIFY(A, i)

Beispiel

Wende BUILD-MAX-HEAP auf das Array
$A = [4, 1, 3, 2, 16, 9, 10, 14, 8, 7]$ an.

Heapsort

HEAPSORT(A)

1 BUILD-MAX-HEAP(A)
2 **for** $i = A.length$ **downto** 2
3 vertausche $A[1]$ mit $A[i]$
4 $A.heap\text{-}size = A.heap\text{-}size - 1$
5 MAX-HEAPIFY($A, 1$)

Beispiel

Sortiere das Array $A = [4, 1, 3, 2, 16, 9, 10, 14, 8, 7]$ mit HEAPSORT.

Prioritätswarteschlangen

Prioritätswarteschlangen (engl. priority queues) lassen sich mit Heaps effizient implementieren. Dazu benötigt man die folgenden Operationen bei einer Max-Priority Queue:

- MAX-HEAP-INSERT(A, x): Füge x in Heap A ein ($O(\log n)$)
- HEAP-MAXIMUM(A): Gebe das Maximum von A zurück ($O(1)$)
- HEAP-EXTRACT-MAX(A): Entferne das Maximum von A ($O(\log n)$)
- HEAP-INCREASE-KEY(A, x, k): Erhöhe den Schlüssel von x auf k ($O(\log n)$)

Aufgabe

Wie lassen sich Queues und Stacks durch eine Priority Queue implementieren?

Aufgaben

Aufgabe

Die Operation HEAP-DELETE(A, i) entfernt ein Element i vom Heap A und erhält dabei die Heap-Eigenschaft. Implementiere HEAP-DELETE in Pseudocode.

Aufgabe d-näre Heaps

Definition

Ein **d-närer Heap** ist eine Verallgemeinerung des binären Heaps mit d statt 2 Kindern.

Aufgaben

1. Wie lässt sich ein d-närer Heap in einem Array repräsentieren?
2. Was ist die Höhe eines d-nären Heaps mit n Elementen?

Übersicht


THE ENTIRE ENGLISH LANGUAGE IS BUILT OUT OF TWENTY-SIX CHARACTERS, ALONG WITH SOME PUNCTUATION.
IT'S AMAZING THAT WE CAN BASICALLY DESCRIBE ALL OF OUR MYRIAD OF EXPERIENCES WITH SUCH A LIMITED NUMBER OF SYMBOLS.
ACTUALLY, IT ONLY REQUIRES TWO SYMBOLS.
calamitiesofnature.com © 2010 Tony Piro