Prof. Andreas Goerdt
A. Lanka

TU Chemnitz
Sommersemester 2004

Theoretische Informatik III

1. Übung

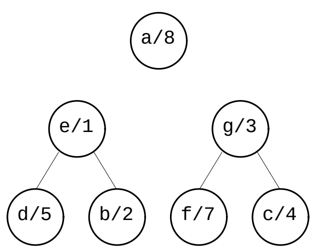
1. Aufgabe:

   Zeigen Sie die Korrektheit des Floyd-Warshall-Algorithmus. Versuchen Sie dazu, die folgende Invariante zu beweisen: „Nach dem k-ten Durchlauf der äußersten Schleife gibt $A[i, j]$ die Länge des kürzesten Weges von i nach j an, der als Zwischenknoten nur die Knoten $1, \dots, k$ benutzt.“

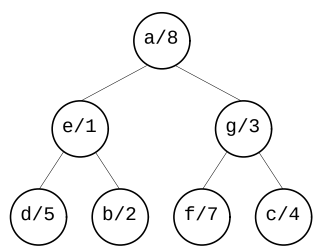
2. Aufgabe:

   Hat man zwei Heaps und einen zusätzlichen Knoten gegeben, läßt sich daraus leicht ein einziger Heap generieren. Man setzt den einzelnen Knoten als Wurzel und hängt daran die beiden Heaps. Damit die Heap-Eigenschaft wieder hergestellt wird, läßt man den Wurzelknoten so lange nach unten sickern, bis ein korrekter Heap entsteht.

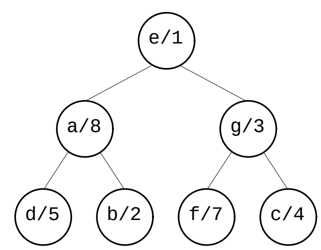
   Beispiel:



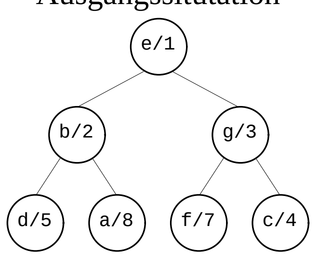
Ausgangssitutation



Zusammenfügen

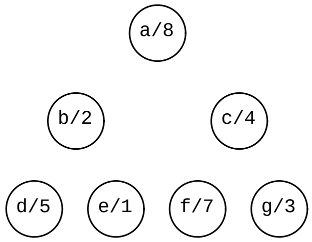


Knoten einmal sickern lassen

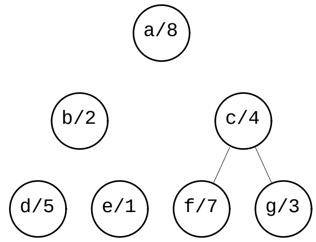


Knoten noch einmal sickern lassen

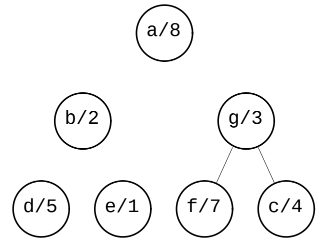
   Diese Methode kann auch benutzt werden, um Heaps aufzubauen. Wir betrachten ein gegebenes Feld $\begin{array}{|c|c|c|c|c|c|c|}\hline 8 & 2 & 4 & 5 & 1 & 7 & 3 \\ \hline\end{array}$, das mit $a, \dots, g$ indiziert ist. Wir benutzen die eben beschriebene Methode, um aus diesem Feld einen korrekten Heap aufzubauen:



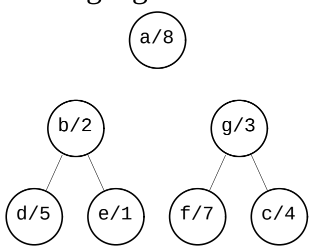
Ausgangssitutation



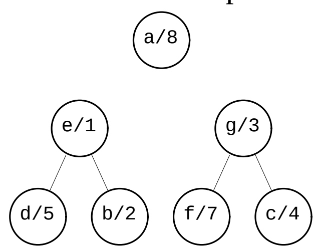
einen kleinen Heap bauen



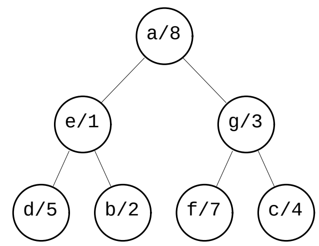
Wurzel sickern lassen



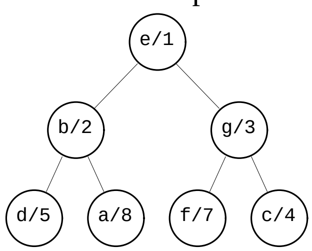
nächsten Heap bauen



dessen Wurzel sickern lassen



einen großen Heap bauen



dessen Wurzel sickern lassen

Das Feld hat am Ende also das Aussehen

1	2	3	5	8	7	4

. Dieses Verfahren arbeitet natürlich nur auf einem Feld. Die dargestellten Bäume sind nur zur Visualisierung.

a) Überlegen Sie sich, wie der Vater und die Söhne eines Heap-Elementes bestimmt werden können.

b) Implementieren Sie eine Sicker-Funktion in Pseudocode.

c) Implementieren Sie das Verfahren zum Heap-Aufbau in Pseudocode. Eingabe ist ein beliebiges Feld der Länge n.

d) Schätzen Sie die Laufzeit Ihrer Implementierung ab.

3. Aufgabe:

Beweisen Sie die folgenden Sätze.

a) Für alle $k, n \in \mathbb{N}$ mit $n, k \geq 1$ ist $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$.

b) Für alle $n \geq 0$ ist $\sum_{i=0}^{n} \binom{n}{i} = 2^n$.

c) Für alle $k, l, n \in \mathbb{N}$, $k \leq l + n$ gilt $\binom{l+n}{k} = \sum_{i=0}^{l} \binom{l}{i} \cdot \binom{n}{k-i}$.

Dabei gilt $\binom{a}{b} = 0$ falls $a < b$ ist.
Hinweis: Arbeiten Sie mit der „kombinatorischen Interpretation“ der Binomialkoeffizienten:

$$\begin{aligned}\binom{l+n}{k} &= \#\, k\text{-elementige Teilmengen einer } (l+n)\text{-elementigen Menge} \\ &= \#\, k\text{-elementige Teilmengen einer Menge,} \\ &\quad\ \text{die aus } l \text{ „roten“ und } n \text{ „schwarzen“ Elementen besteht.}\end{aligned}$$