Prof. Dr. A. Goerdt
J. Pape-Lange

TU Chemnitz
Sommersemester 2021
27.04.2021

# Effiziente Algorithmen/ Theoretische Informatik III

## 1. Übung

**1. Aufgabe:**
Wir betrachten den *Heap* aus der Vorlesung.
Wie viele Knoten kann der Heap mit Tiefe $t$ minmal haben?
Wie viele Knoten kann der Heap mit Tiefe $t$ maximal haben?
Was ist der Index des am weitesten links stehenden Knoten mit Tiefe $t$?

**2. Aufgabe:**
Wir betrachten den *Heap* aus der Vorlesung.
Der *MinimumLöschen*-Algorithmus aus der Vorlesung funktioniert so:

- Tausche das erste Element des Heaps und das letzte Element des Heaps.
- Tausche das nach vorne getauschte letzte Element des Heaps mit dem kleineren Kind, bis die Heapbedingung wieder hergestellt ist.

Geben Sie die minimale Anzahl der Vertauschungen in Abhängigkeit von der Größe $n$ des Heaps an!
Folgender Algorithmus funktioniert nicht:

- Lösche den Wurzel-Knoten.
- Rücke das kleinste Kind der Wurzel nach oben. Die anderen Knoten bleiben dabei an ihrem Platz, sodass an der Stelle des Kindes ein freie Stelle ohne Knoten entsteht.
- Solange es eine Stelle mit Kindern aber ohne Knoten gibt, rücke das kleinste Kind dieser Stelle nach oben.

Erklären Sie, warum dieser Algorithmus nicht funktioniert!

**3. Aufgabe:**
Implementieren Sie einen *Heap* in einer Programmiersprache Ihrer Wahl.

**4. Aufgabe:**
Wir betrachten *Dijkstras Algorithmus* für *kürzeste Wege* in gerichteten Graphen mit *nicht-negativen* Kantengewichten.

(a) Wiederholen Sie die Funktionsweise des Algorithmus!

(b) Welche Laufzeiten ergeben sich mit verschiedenen Datenstrukturen (*Array*, *Heap*) zur Verwaltung der aktuellen Suchfront?

(c) (schwer): Überlegen Sie sich, warum der *Dijkstra-Algorithmus*, unabhängig von der verwendeten Datenstruktur, im Allgemeinen keine Laufzeit besser als $\mathcal{O}(|V| \cdot \log |V|)$ erreichen kann!