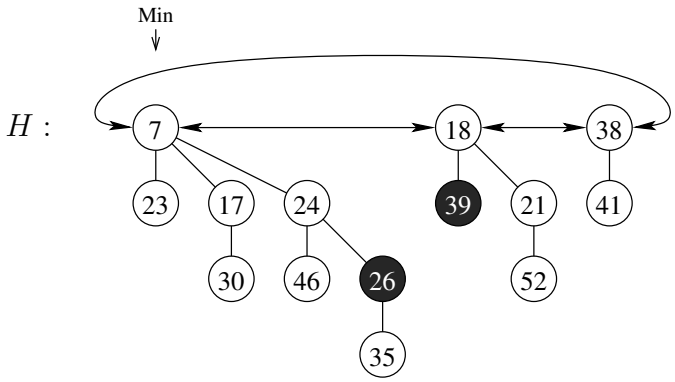
Prof. Dr. A. Goerdt
Julian Pape-Lange

TU Chemnitz
Sommersemester 2021
27.05.2021

Effiziente Algorithmen / Theoretische Informatik III

3. Übung

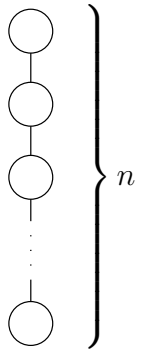
1. Aufgabe: Führen Sie `LöscheMinimum()` auf folgendem Fibonacci-Heap H aus:



Verbessern Sie anschließend die 35 unter der markierten 26 zu 2.

2. Aufgabe:

(a) Geben Sie für jedes $n > 0$ eine Folge von Operationen für den Fibonacci-Heap an, so dass ein Baum der Art



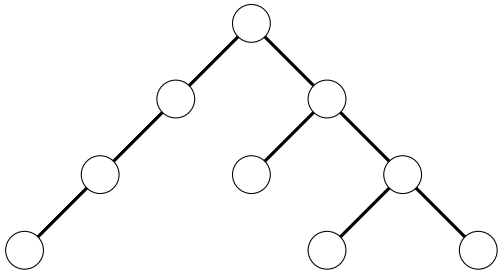
mit n Knoten entsteht.

(b) Zeigen Sie, dass die Operationen `LöscheMinimum()` und `VerbessereSchlüssel(`i`, `j`, `H`)` im Fibonacci-Heap H mit n Knoten eine *worst-case-Laufzeit* von $\Theta(n)$ haben.

(c) Überlegen Sie sich, wie der Fibonacci-Heap auf der Knotenmenge $\{1, 2, 3, \ldots, n\}$ mit nur einem Array dargestellt werden kann, in welchem die Einträge für die jeweiligen Knoten stehen.

3. Aufgabe:
Wir betrachten die *Union-by-Size*-Heuristik.

- Zeigen Sie, dass in allen Teilbäumen gilt, dass die Tiefe durch den Logarithmus der Knoten begrenzt ist.
- Zeigen Sie, dass der folgende Baum nicht durch Union-Operationen mit der Union-by-Size-Heuristik gebildet werden kann.



- Kann jeder Baum, der die im ersten Aufgabenteil gezeigte Aussage erfüllt, durch Union-Operationen mit der Union-by-Size-Heuristik gebildet werden?

4. Aufgabe:
Wir betrachten die Laufzeitabschätzung von Union-Find mit Union-by-Size und Wegkompression. Wir nehmen an, dass vor jeder Union-Operation die beiden Wurzeln mit Find gefunden werden müssen.
Zeigen Sie, dass wir mit dieser Annahme die Abschätzung $\mathcal{O}(N + (M + n)\log^* n)$ zu $\mathcal{O}(M \log^* n)$, $\mathcal{O}(M + n \log^* n)$ und $\mathcal{O}(M \log^* M)$ verbessern können.
Hinweis: Wenn $M < n$ gilt, müssen einige Mengen einelementig sein. Wenn $M > n$ gilt, müssen die Niveauübergänge Elementweise gezählt werden, um auf die zweite Abschätzung zu kommen.

5. Aufgabe:
Vergleichen Sie tabellarisch das Wachstum der Funktionen $log^* n$, $log(log(n))$, $log(n)$, n und $n \log n$.