

Erläuterungen und Ergänzungen zum 2 Meilenstein: Fibonacci-Heaps

1. Vorbereitungen zum Verständnis der Fibonacci-Heaps

Definieren Sie die Funktionen des ADT Fibonacci-Heap und erklären Sie das Prinzip der amortisierten Laufzeit.

Konstruieren Sie auf Papier einen Fibonacci-MinHeap durch sukzessives Hinzufügen der folgenden Prioritäten von links nach rechts und erklären Sie Ihre Vorgehensweise.

3, 19, 21, 4, 6, 1, 17, 33, 9, 18, 29, 24, 59, 5, 25, 36, 2

Wenden Sie auf den Fibonacci-MinHeap die Funktion `DELETEMIN` an und zeichnen Sie den neuen Fibonacci-Heap.

Verringern Sie in dem Fibonacci-Heap Prioritäten und wenden Sie dabei die Heuristik "Pfadverkürzung" an. Achten Sie auf die Markierung der Knoten.

1. $6 \rightarrow 20$
2. $21 \rightarrow 28$
3. $24 \rightarrow 1$
4. $59 \rightarrow 58$
5. $19 \rightarrow 18$

2. Implementation der Klasse Fibonacci-MinHeap

Implementieren Sie die **Klasse FIBONACCIHEAP als Max- und als MinHeap**. Achten Sie darauf, dass die ADT-Operationen in anderen Anwendungen problemlos verwendet werden können. Beliebige Objekte sollen in einer Priority-Queue verwaltet werden (vgl. Meilenstein III: Path-Finding und Min-Cost-Spanning-Tree-Algorithmen).

Programmieren Sie die ADT-Operationen nach dem in der Vorlesung besprochenen Pseudocode, insbesondere die Konsolidierung nach einer `DELETEMIN`-Operation, dort bezeichnet als `FIB-HEAP-EXTRACT-MIN`.

Entwickeln Sie eine **Testumgebung**, um die Korrektheit der ADT-Operationen und deren Performance zu überprüfen:

- Erzeugen eines Fibonacci-MinHeaps aus einer zufälligen Folge von n Prioritäten. Ausführen der Operation `FIB-HEAP-INSERT`.
- Ausführen der `DELETEMIN`-Operation und Ausgabe der Höhe und Anzahl von Knoten des Fibonacci-Heaps vor und nach der Operation.

- Hinzufügen von Prioritäten: im Wechsel n FIB-HEAP-INSERT-Operationen und eine DELETEMIN-Operation. Stellen Sie dar, wie sich die Struktur des Heaps dadurch jeweils verändert.
- Verändern von Prioritäten durch Aufrufen der Operation DECREASE-PRIORITY mit und ohne die Heuristik "Pfadverkürzung - Vermeidung dünner Bäume". Stellen Sie dar, wie sich die Höhe/Struktur des Heaps verändert, je nachdem ob die Heuristik ein- oder ausgeschaltet ist.
- **Experimente mit DELETEMIN.** Erzeugen Sie mit einem Zufallsgenerator n Prioritäten und fügen Sie diese mit FIB-HEAP-INSERT nacheinander in einen Fibonacci-Heap ein. Führen Sie danach DELETEMIN aus und messen Sie die Laufzeit von DELETEMIN. Führen Sie ähnliche Experimente wie zum Meilenstein I durch. Vergrößern Sie also die Anzahl der Prioritäten n , fügen n Prioritäten in den Fib-Heap ein und führen danach DELETEMIN aus. Stellen Sie mit Ihrem Programm zur experimentellen Laufzeitmessung die Ergebnisse dar. Verwenden Sie den Powertest und zeichnen Sie die Grafiken und die Laufzeitfunktion $T_{\text{DELETEMIN}}(n)$ zu den gemessenen Laufzeiten von DELETEMIN. Vergleichen Sie die Ergebnisse mit den Werten der theoretischen Analyse (Vorlesung Kap. 5.2 Folie 3).

3. Präsentation der Ergebnisse zum Meilenstein-II-Termin

Wie schon beim ersten Meilenstein, ist eine Präsentation und eine Dokumentation vorzubereiten, mit der Sie das Konzept und die Eigenschaften von Fibonacci-Heaps sowie die Ergebnisse der Tests mit der Testumgebung erklären. Anschließend führen Sie die ADT-Funktionen und die Experimente mit dem Testprogramm vor.

Schicken Sie bitte 2 Tage vor dem Meilensteintermin die Dokumentation im PDF-Format an alex.maier@fh-koeln.de und heinrich.klocke@fh-koeln.de mit dem **Betreff** „<Teamname>_MS_<Meilensteinnummer>“.