**Algorithmik-Praktikum**

**Informatik**

**Wintersemester 2019/2020**

**Binär- Heaps und Binomial- Heaps**

Teilnehmer (Luca Stamos, Stefan Steinhauer, Dimitri Osokin, Marc Kevin Zenzen)

**Inhaltsverzeichnis**

**Einleitung**

Binär- und Binomial Heaps sind Datenstrukturen, welche den abstrakten Datentyp Prioritätswarteschlangen implementieren. Das bedeutet, dass jedem einzelnen Element des Heaps eine Priorität zugewiesen wird. Im nachfolgenden werden wir das notwendige Grundwissen behandeln und uns anschließend tiefer mit den einzelnen Teilthemen beschäftigen. Dabei gehen wir auf die verschiedenen Operationen der Heaps ein, wie z.B. Insert (Element hinzufügen), ExtractMin (entnimmt das Element mit dem niedrigsten Schlüssel), HeapifyUp (sortiert ein Element von unten nach oben in den Tree ein)/HeapifyDown (sortiert ein Element von oben nach unten in den Tree ein) oder Merge (Teilbäume zusammenfügen). Die Laufzeit der Heaps spielt hierbei auch eine entscheidende Rolle, da die Effektivität des Heapsn unter anderem durch sie bestimmt wird. Zuletzt überlegen wir, wie wir die Datenstrukturen in einem praxisrelevanten Beispiel implementieren können.

**Recherche / Grundlagen**

Bevor wir auf die Binär-Heaps und Binomial-Heaps zu sprechen kommen, gehen wir auf Baumstrukturen ein, da diese die Grundlagen für Heaps darstellen.

**Definition der Lernziele**

Sie verstehen den Sinn von Heap-Strukturen und dessen Einsatz in der Programmierwelt. Sie können die verschiedenen Funktionen eines Heaps verstehen und implementieren.=

**Thema**

In folgendem Abschnitt gehen wir genauer auf die beiden Themen „Binär-Heaps“ und „Binomial-Heaps“ ein. Wir werden hier genauer auf die Verwendung der Prioritätswarteschlangen und auf die Unterschiede zwischen den Heaps eingehen. Wo die Vor- und Nachteile der Heaps liegen oder wo Sie sich zu anderen Sortieralgorithmen unterscheiden.

Binär-Heaps:

Binomial-Heaps:

Bei den Binomial-Heaps werden auch die Operationen insert, und extractMin behandelt, zusätzlich kommt hier aber noch die Operation merge hinzu. Mit dieser werden einzelne Teilbäume zusammengehängt.

Laufzeitvergleich:  
// Binär vs Binomial vs andere Heaps

// Suchalgorithmen vs Heapsort

**Implementierung**

**Literaturliste**

Th. H. Cormen: Algorithmen - Eine Einführung Oldenburg 2. + 4. Auflage

Jean Vuillemin - A Data Structure for Manipulating Priority Queues

<https://de.wikipedia.org/wiki/Bin%C3%A4rer_Heap#Siehe_auch><https://de.wikipedia.org/wiki/Binomial-Heap>

**Literaturverzeichnis**