

---

## Effiziente Algorithmen und Datenstrukturen I

---

*Abgabetermin: 31.01.2005 vor der Vorlesung*

### **Aufgabe 1**

Zeigen Sie, wie man unter  $n$  Elementen mit Hilfe eines Soft-Heaps in  $O(n)$  Schritten ein Element finden kann, dessen Rang für ein vorgegebenes  $\varepsilon$  um maximal  $\varepsilon n$  von  $\lfloor \frac{n}{2} \rfloor$  abweicht, wobei  $0 < \varepsilon \leq \frac{1}{2}$  gilt.

### **Aufgabe 2**

Wie muss man den Parameter  $\varepsilon$  wählen, damit ein Soft-Heap eine Binomial-Queue ist? Bestimmen Sie für diesen Fall die Laufzeiten aller Priority-Queue-Operationen.

### **Aufgabe 3**

Zeigen Sie, dass die Höhe eines Fibonacci-Heaps nicht durch  $O(\log n)$  beschränkt werden kann, indem Sie zeigen, dass es zu jeder Zahl  $n$  eine Folge von Fibonacci-Heap-Operationen gibt, die einen Fibonacci-Heap erzeugt, der nur aus einem linear entarteten Baum mit  $n$  Schlüsseln besteht.

### **Aufgabe 4**

Zeigen Sie, dass jede Sequenz von  $m$  MAKE-SET, FIND und UNION-Operationen, in der alle UNION-Operationen vor der ersten FIND-Operation kommen, nur  $O(m)$  Schritte benötigt, falls sowohl die *Path-Compression*-Heuristik, als auch die *Union-by-Rank*-Heuristik verwendet werden.