Prof. Dr. A. Goerdt
L. Falke

TU Chemnitz
Sommersemester 2015
22.05.2015

Effiziente Algorithmen / Theoretische Informatik III

4. Übung

1. Aufgabe: Wir untersuchen, wie sich der Splaybaum auf dem Anfangsbeispiel der Vorlesung verhält:
Fügen Sie die Elemente $n, n-1, n-2, ..., 1$ in einen anfänglich leeren Splaybaum ein. Führen Sie danach `Find`(n) aus. Überlegen Sie sich, dass die Kosten der zugehörigen Folge von Operationen tatsächlich

$$O((n+1) \cdot \log\ n)$$

betragen. Überlegen Sie sich, wie die darauffolgenden `Find`$(n-1)$, ... , `Find`(1) bearbeitet werden. Wie kann danach wieder ein „ganz dünner“ Baum entstehen?

2. Aufgabe: Angenommen, durch eine Folge von Operationen auf einem anfangs leeren Splaybaum entsteht ein idealer, das heißt vollständig ausgeglichener binärer Suchbaum. Welches Potential Φ im Sinne der amortiserten Analyse besitzt dieser Baum?

3. Aufgabe: Zeigen Sie die amortisierten Laufzeiten für die Splaybaum-Operationen `Insert(`x`)` und `Delete(`x`)`.

Hinweis: Benutzen Sie, dass ein `Find(`x`)` amortisiert höchstens $3 \log n + 4$ dauert.

4. Aufgabe: Gegeben seien Objekte mit Schlüsselwerten $w_1 < w_2 < \ldots < w_n$ mit den zugehörigen relativen häufuigkeiten $p_1, p_2, \ldots, p_n$ des Zugriffs auf die entsprechenden Schlüssel. Gesucht ist ein Verfahren zur Konstruktion eines *optimalen statischen binären Suchbaums* zu den w_i.
Das ist ein Suchbaum bei dem die Laufzeit, über eine (unendlich) lange Folge von `Find`-Operationen gesehen, möglichst klein ist.