Das *Letzte* vor der Klausur

Zum organisatorischen Ablauf

- Die Klausur findet wie geplant am Freitag, den 17.6.92 in den drei Hörsälen der Physik statt.
- Klausurrelevant ist der gesamte Stoff einschließlich des Abschnitts *balancierte Bäume*, also auch die 9. Übung.
- Es wird keine Aufgaben geben, die bearbeitet werden *müssen*. Alle Aufgaben sind gleichberechtigt, um die erforderliche Punktzahl zu erreichen.
- Programmieraufgaben müssen mittels korrektem *PASCAL*–Sourcecode bearbeitet werden (keine Ausnahme!).
- Es werden Listen ausgehängt, auf denen die Zulassung zur Klausur und die Zuordnung zu einem der drei Hörsäle entnommen werden kann.

Inoffizielle Tips zur Klausur

- Alle Übungsaufgaben anschauen, am besten nachrechnen!
- Vorlesung anschauen und alle *Ideen* verstehen. Dazu gehören alle Datenstrukturen, Algorithmen mit Laufzeiten und Definitionen.
- Folgende Begriffe sollten auf jeden Fall bekannt sein:
  - Definitionen der asymptotischen Laufzeiten $O(f), \Theta(f), \Omega(f)$, Einordnung von gegebenen Funktionen in Laufzeitklassen
  - Begriffe worst–case, average–case, best–case, Gleichverteilung der Daten bei average–case
  - Datenstrukturen wie Liste (einfach oder doppelt verkettet, mit oder ohne Dummys), Array, Stack, Queue, binärer Baum, jeweils Vor– und Nachteile herausstellen, Laufzeiten für Standard–Operationen wie Search, Insert, Delete, Ord, Predecessor, Successor, Concatenate (nicht alle werden auf allen Datenstrukturen unterstützt)
  - Sortierverfahren, Algorithmen, Unterschiede, Laufzeiten, Speicherbedarf
  - Auswahlproblem (SELECT), Algorithmus, wodurch wird Laufzeit $O(n)$ erreicht?
  - Suchverfahren, Algorithmen, Laufzeiten, Unterschiede, Voraussetzungen
  - Hashverfahren, Verkettung von Überläufern, offene Verfahren, Sondierungsmethoden
  - Binäre Bäume, speziell Rot–Schwarz–Bäume, AVL–Bäume, Balancierung, Rotationen

Viel Erfolg!

Zur Übung

Hierbei handelt sich es um Übungsaufgaben der Informatik I SS 1990.

Aufgabe 1

Verwende folgende rekursive Methode zur Berechnung des Maximums und des Minimums einer n–elementigen Menge S.

1. Partioniere S in S_1 und S_2 mit $|S_1| = \lceil |S|/2 \rceil$, $|S_2| = \lfloor |S|/2 \rfloor$.
2. Berechne rekursiv $Max(S_1)$, $Min(S_1)$, $Max(S_2)$, $Min(S_2)$.
3. Berechne daraus $Max(S)$ und $Min(S)$.

Sei $T(n)$ die Anzahl der Vergleiche, die der obige Algorithmus durchführt, um in einer n–elementigen Menge Maximum und Minimum zu finden. Zeige durch vollständige Induktion über $k > 0$ speziell für $n = 2^k$:

$$T(n) = \frac{3}{2}n - 2$$

Aufgabe 2

In jedem Knoten eines binären Suchbaums T sei zusätzlich die Anzahl der Elemente im linken Teilbaum abgespeichert. Gib einen Algorithmus an, der möglichst schnell die Operation Ord(k) durchführt und analysiere die Laufzeit.

Definition

Ein *rang–balancierter Baum* ist ein binären Baum, bei dem für jeden Knoten v eine ganze Zahl $Rang(v)$ mit folgenden Eigenschaften gegeben ist:

1. $Rang(v) \leq Rang(Vater(v)) \leq Rang(v) + 1$
2. $Rang(v) < Rang(Vater(Vater(v)))$
3. Ist v ein Blatt, so ist der $Rang(v) = 0$.

(Tüftel–)Aufgabe 3

Zeige daß für jeden binären Baum T gilt:

$$\text{T ist rangbalanciert} \quad \Longleftrightarrow \quad \text{T ist Rot–Schwarz–Baum}$$