

5.1 a-b-Bäume (12 Punkte)

- (2P) Welche Probleme ergeben sich, wenn $2 \leq a \leq \frac{b+1}{2}$ nicht erfüllt ist?
- (1P) Kann man die "Stehlen"-Operation weglassen und immer fusionieren, ohne Laufzeit einzubüßen?
- (1P) Wie ändert sich die erwartete Laufzeit beim Suchen im Baum, wenn innere Knoten keine Schlüssel speichern, sondern diese bei Bedarf immer neu ermitteln?
- In der Vorlesung wurde das Einfügen und Löschen in einem a-b-Baum vorgestellt. Laden Sie sich den Beispielcode des 2-5-Baums von der Homepage herunter und erledigen Sie folgende Aufgaben:
 - (1P) Zeichnen Sie eine komplette Grafik des Baumes nach dem Einfügen der Werte $\{1,2,3,4,5,6\}$, in der alle Instanzen der Klassen aus `ABTree.java` vorkommen, sowie deren Referenzen aufeinander.
 - (1P) Kommentieren Sie den Quellcode.
 - (3P) Ergänzen Sie die Methode `split` und `fusion` in `ABTreeInnerNode`.¹
 - (3P) Ergänzen Sie die Methoden `insert` und `remove` in `ABTreeRootNode`.²
 - (3 Bonuspunkte) Verbessern Sie das Programm, indem Sie es so modifizieren, dass innere Knoten Schlüssel speichern und diese nur aktualisieren, wenn dies nötig ist.

Den Test in SAUCE besteht ihr Programm nach Implementierung von `split`, `fusion`, `insert` und `remove`.

5.2 Amortisierte Analyse dynamischer Vektoren (4 Punkte)

Gegeben sei ein dynamischer Vektor³. Durch die Operation `add` kann ein neues Element an das Ende des Vektors hinzugefügt werden. Intern sei der Vektor durch ein statisches Feld realisiert, dessen Größe sich bei Bedarf verdoppelt.

- (1P) Berechnen Sie die worst-case Kosten für das Hinzufügen eines Elementes zu einem Vektor mit n Einträgen.
- (3P) Zeigen Sie, dass die operation `add` konstante amortisierte Kosten besitzt, indem Sie folgendes Kontoführungsschema benutzen:

$$\text{Konto}(i) = \text{size}(i) - \text{capacity}(i).$$

Dabei ist $\text{size}(i)$ die Anzahl der gespeicherten Elemente und $\text{capacity}(i)$ die Größe des internen Feldes.

¹Achten Sie dabei darauf, dass der Knoten, der die Methode aufruft, im Baum erhalten bleibt. D.h. `split` fügt nur einen weiteren inneren Knoten hinzu und ersetzt nicht den aktuellen durch zwei neue. Genauso soll nach einem Aufruf von `fusion` der aktuelle Knoten erhalten bleiben und ein Geschwisterknoten entfernt werden und nicht beide Knoten entfernt und durch einen neuen ersetzt werden.

²Behandeln Sie darin gesondert das Verhalten des Wurzelknotens, falls ein `split` oder `fusion` stattgefunden hat.

³wie z.B. `ArrayList` in Java

5.3 Erzeugende Funktionen (4 Punkte)

- a) (3P) Lösen Sie folgende lineare Rekursionsgleichung mit Hilfe erzeugender Funktionen

$$L_0 := 2$$

$$L_1 := 1$$

$$L_n := L_{n-1} + L_{n-2} \text{ für } n \geq 2$$

und zeigen Sie

$$L(z) = \frac{2-z}{1-z-z^2} = \frac{1}{1-\Phi z} + \frac{1}{1-\bar{\Phi} z}.$$

- b) (1P) Zeigen Sie folgenden Zusammenhang mit den Fibonacci-Zahlen

$$L_n = F_{n-1} + F_{n+1} \text{ für } n \geq 1.$$