



1. (15 Punkte) Wir haben gesehen, dass in einem binären Suchbaum die Operationen SUCHEN, EINFÜGEN, und LÖSCHEN Laufzeit $O(h)$ haben, wobei h die Höhe des binären Suchbaums ist (also, h ist die Anzahl der Kanten auf einem längsten einfachen Wurzel-Blatt Pfades).
 - (a) Was ist die maximale Höhe eines binären Suchbaums mit n Knoten?
 - (b) Was ist die minimale Höhe eines binären Suchbaums mit n Knoten?
2. (20 Punkte) Zusätzlich zu den schon aus der Vorlesung bekannten Operationen auf einem binären Suchbaum (BST) sollen nun auch die Operationen $SUCC(x)$ (zum Finden des Nachfolgers von x) und $PRED(x)$ (zum Finden des Vorgängers von x) unterstützt werden. Dabei ist der Nachfolger von x das nächstgrößte Element nach x und der Vorgänger das nächstkleinere Element von x . Implementieren Sie diese Operationen auf einem BST. Was sind die Laufzeiten Ihrer Algorithmen? Gehen Sie dabei davon aus, dass x durch einen Zeiger auf den Baumknoten gegeben ist, in dem x gespeichert ist.
Hinweis: Sie können Elternzeiger benutzen.
3. (25 Punkte) In der Vorlesung haben wir über das Problem der Auswahl nach Rang gesprochen: Gegeben sei eine Menge X mit n Schlüsseln und eine Zahl k mit $1 \leq k \leq n$. Wir möchten den k -kleinsten Schlüssel in X finden. Wir haben gesehen, dass dies mit $O(n)$ Laufzeit möglich ist. Im Folgenden sollen Sie zeigen, dass sich dieses Problem schneller lösen lässt, wenn die Elemente in einem Binären Suchbaum gespeichert sind.
 - (a) Nehmen Sie an, Sie dürfen einen BST konstruieren, der X repräsentiert. Finden Sie eine Methode, um mit Hilfe dieses BST das Auswahlproblem Zeit in $O(h)$ zu lösen, wobei h die Höhe des BST ist (die Laufzeit beinhaltet nicht die Konstruktion des BST). *Hinweis:* Sie können zusätzliche Daten in den Knoten speichern.
 - (b) Nun sollen Änderungen von X durch Einfüge- und Löschoptionen ermöglicht werden. Wie müssen Sie die insert und delete-Operationen für BSTs anpassen, sodass Ihr Algorithmus aus Teil a) immer noch funktioniert? Die Operationen sollen immer noch $O(h)$ Laufzeit haben.