
Grundlagen: Algorithmen und Datenstrukturen

Abgabetermin: Jeweilige Tutorübung in der Woche vom 25. bis 29. Juni

Tutoraufgabe 1

Zeigen Sie, dass jeder Binärbaum mit b Blättern genau $b - 1$ innere Knoten besitzt, wenn jeder innere Knoten genau zwei Kinder besitzt. Ein Knoten eines gewurzelten Baumes ist genau dann ein Blatt, wenn er keine Kinder besitzt.

Anmerkung: In der Literatur wird ein Baum, der nur aus der Wurzel besteht, nicht einheitlich gehandhabt - in einigen Fällen ist dieser einzelne Knoten per Definition ein innerer Knoten, in anderen Fällen ein Blatt, und in wieder anderer Literatur ein Spezialfall, der weder innerer Knoten, noch Blatt ist. Bei unserer Definition ist ein solcher Knoten ein Blatt.

Tutoraufgabe 2

Zeigen Sie die für die Vorlesung sowie die nächsten Tutoraufgaben benötigte Aussage:

Sei T ein fast vollständiger Binärbaum mit $n \geq 1$ Knoten und Baumtiefe $t \geq 0$. Dann gilt

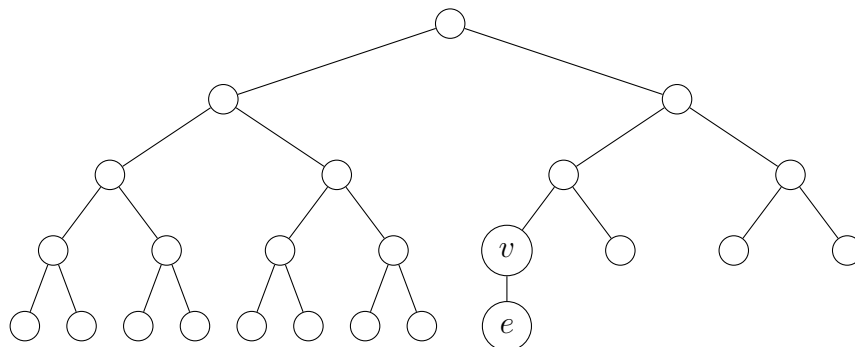
$$2^t \leq n \leq 2^{t+1} - 1.$$

Definitionen: Die Tiefe eines Knotens ist die Länge des (eindeutigen) Pfades von dem Knoten zur Wurzel, gemessen in der Zahl der Kanten des Pfades. D.h. insbesondere, dass die Wurzel Tiefe 0 hat. (Dies ist nicht einheitlich in der Literatur: in einigen Fällen wird in der Zahl der Knoten des Pfades gemessen, was dann gerade 1 mehr ist als die Zahl der Kanten). Die Baumtiefe eines gewurzelten Baumes ist das Maximum der Tiefen aller enthaltenen Knoten. Das i -te Level eines gewurzelten Baumes besteht aus allen Knoten, die Tiefe i besitzen. Ein Binärbaum ist vollständig, wenn jeder innere Knoten genau zwei Kinder hat und alle Blätter dieselbe Tiefe haben.

Sei t die Tiefe eines Binärbaumes T . Der Baum T ist ein fast vollständiger Binärbaum, wenn T entweder nur aus der Wurzel besteht, oder wenn die ersten $t - 1$ Level gemeinsam einen vollständigen Binärbaum bilden, und die Knoten von T so angeordnet werden können, dass es auf dem Level t einen Knoten e mit dem Vaterknoten v gibt, sodass gilt:

- Alle Knoten auf dem Level $t - 1$, die „links“ von v stehen, haben zwei Kinder.
- Alle Knoten auf dem Level $t - 1$, die „rechts“ von v stehen, haben keine Kinder.

Die folgende Abbildung zeigt einen fast vollständigen Binärbaum mit Baumtiefe 4.



Tutoraufgabe 3

Zeigen Sie, dass es für jedes $b \geq 1$ einen fast vollständigen Binärbaum mit b Blättern gibt, sodass jeder innere Knoten genau zwei Kinder hat.

Tutoraufgabe 4

In der Vorlesung wurde der QuickSelect-Algorithmus vorgestellt, der bei zufälliger (gleichverteilter) Eingabe erwartet lineare Laufzeit besitzt, jedoch im Worst-Case-Laufzeit quadratischen Aufwand hat. Es gibt deterministische Selektionsverfahren, die auch im Worst-Case das gesuchte Element in linearer Laufzeit finden. Eines davon ist der BFPRT-Selektionsalgorithmus (Blum-Floyd-Pratt-Rivest-Tarjan), der vom Wesen her ein modifizierter QuickSelect-Algorithmus ist, der die Pivot-Elemente geschickt auswählt. Die Konstante bezüglich der asymptotischen oberen Schranke des BFPRT-Algorithmus ist allerdings „relativ“ groß (jedoch maximal etwa 20, siehe Skript der Vorlesung Effiziente Algorithmen und Datenstrukturen von WS 2010).

Wir betrachten in dieser Aufgabe ein eingeschränktes Problem, nämlich dasjenige, das *zweitgrößte* Element einer Zahlenfolge der Länge n zu finden. Hierbei nehmen wir an, dass keine Zahl doppelt in der Zahlenfolge vorkommen kann. Prinzipiell würden dafür maximal $3n$ Vergleiche genügen, um das größte und zweitgrößte Element auf vollkommen triviale Weise zu ermitteln. Beschreiben Sie einen Algorithmus, der dieses Problem mit maximal $n + \lg n - 1$ Vergleichen zwischen den Zahlen der Eingabefolge löst.

Hausaufgabe 1

Implementieren Sie den QuickSelect-Algorithmus.

Implementieren Sie in der Klasse `UISqsArray` in der Funktion `quickSelect` den QuickSelect-Algorithmus der das i -te Elemente in dem unsortierten Feld `A` liefert.

Verwenden Sie für Ihre Implementierung die auf der Übungswebseite bereitgestellten Klassen und verändern Sie für Ihre Implementierung *ausschließlich* die Klasse `UISqsArray`.

Achten Sie bei der Abgabe Ihrer Aufgabe darauf, dass Ihre Klasse `UISqsArray` heißt und auf den Rechnern der Rayhalle (rayhalle.informatik.tu-muenchen.de) mit der bereitgestellten Datei `main_qs` kompiliert werden kann. Anderenfalls kann eine Korrektur nicht garantiert werden. Achten Sie darauf, dass Ihr Quelltext ausreichend kommentiert ist.

Schicken Sie die Lösung per Email mit dem Betreff `[GAD] Gruppe <Gruppennummer>` an Ihren Tutor.

Hausaufgabe 2

Veranschaulichen Sie das Vorgehen des RadixSort-Algorithmus anhand der folgenden Wörter:

alt, hort, kalt, bar, sport, spalt, ort, stall, spart, speer, bart

Benutzen Sie die lexikographische Sortierung (Dabei sei das Leerzeichen \sqcup kleiner als jeder andere Buchstabe). Es reicht aus, wenn sie als Alphabet die Buchstaben \sqcup , a, b, e, h, k, l, o, p, r, s, t verwenden und daher die Schlüssel $0, \dots, 11$ verwenden.

Hausaufgabe 3

Zeigen Sie, dass jeder *gewurzelte* Baum mit b Blättern maximal $b-1$ innere Knoten besitzt, wenn jeder innere Knoten mindestens zwei Kinder besitzt. Ein Knoten eines gewurzelten Baumes ist genau dann ein Blatt, wenn er keine Kinder besitzt.