

Algorithmik-Praktikum SS 2004

Aufgabe 4: Zeichnen von binären Bäumen mit dem Divide-and-Conquer-Verfahren *Layered-Tree-Draw*

Aufgabe 5: Anwenden des in Aufgabe 4 entwickelten Programms auf AVL-Bäume

Name: Matr-Nr:

Datum: Unterschrift des Dozenten (wenn bestanden):

Ziel dieser Aufgabe ist es, einen Divide-and-Conquer-Algorithmus zu entwerfen und zu implementieren, der Bäume in einer ästhetisch ansprechenden Form auf dem Bildschirm zeichnet.

Aufgabe 4

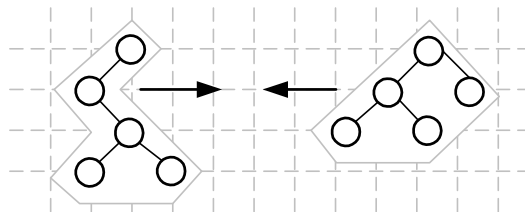
Der in der Vorlesung besprochene Algorithmus *Layered-Tree-Draw* stellt einen binären Baum als Ebenen-Zeichnung dar. Die induktive Idee des D&C-Algorithmus ist folgende:

Algorithmus *Layered-Tree-Draw*

Input: ein binärer Baum T

Output: eine Ebenen-Zeichnung von T

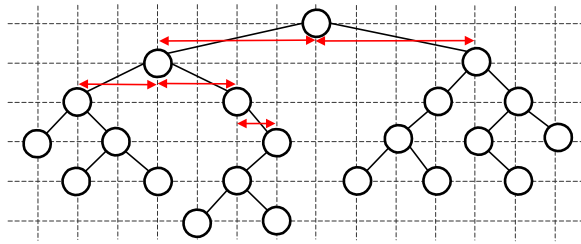
1. Basis-Fall: Wenn T nur aus einem Knoten besteht, dann ist die Zeichnung trivial definiert
2. Divide: Wende den Algorithmus rekursiv auf den linken und rechten Teilbaum von T an
3. Conquer: Wir stellen uns vor, dass jeder Teilbaum auf einem eigenen Blatt Papier gezeichnet wird. Schiebe die Zeichnungen zweier Teilbäume solange horizontal aufeinander zu, bis der horizontale Abstand mindestens 2 wird:



Platziere die Wurzel r von T vertikal 1 Einheit über und horizontal mittig zwischen den Kindern. Falls die Wurzel nicht mittig platziert werden kann, dann müssen die beiden Teilbäume den Abstand 3 haben. Hat r nur einen Teilbaum, z.B. den linken, dann platziere r in horizontalem Abstand 1 rechts vom linken Kind.

Die folgende Abbildung zeigt einen Baum, der die im Conquer-Schritt aufgeführten Bedingungen erfüllt.

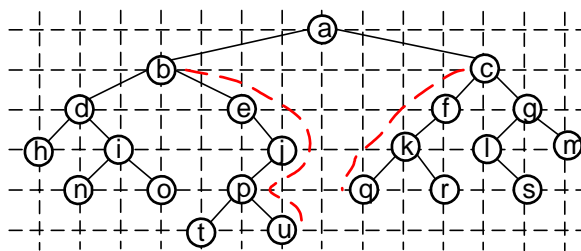
Im Vorlesungsskript ist erklärt, wie der Conquer-Schritt konzeptionell zu realisieren ist.



Jeder Knoten dieses Baumes hat auf jeder (horizontalen) Ebene den Mindestabstand 2 oder 3.

Berechnen Sie zunächst mit einem Postorder-Scan (siehe Vorlesung) für jeden Knoten v den horizontalen Platzbedarf des rechten und linken Sohnes von v bezüglich v . Das Bild zeigt, dass der linke Sohn der obersten Wurzel v hat einen horizontalen Platzbedarf bezüglich v von 4 Einheiten; das gleich gilt für den rechten Sohn.

Die im Postorder-Scan berechneten Werte werden benutzt, um für den Baum mit Wurzel v einen Linksumriss des rechten und einen Rechtsumriss des linken Teilbaumes zu berechnen:



Rechtsumriss des linken Teilbaumes: $b - e - j - p - u$

Linksumriss des rechten Teilbaumes: $c - f - k - q$

Die Verfahren zur Berechnung der Links- und Rechtsumrisse sind im Vorlesungsskript in Kapitel 12.10.1 erklärt. Durch die Umriss-Listen sind die x -Koordinaten für jeden Knoten bekannt. Die y -Koordinaten ergeben sich aus dem Tiefenlevel der Knoten. Damit kann jeder Knoten an seiner x - und y -Koordinate gezeichnet werden.

Was müssen Sie tun?

- Recherchieren Sie nach einer geeigneten Klassenbibliothek in Java, mit der binäre Suchbäume repräsentiert werden können.
- Entwerfen und Programmieren Sie den *Layered-Tree-Draw-Algorithmus*, um einen "beliebig" großen Suchbaum zu zeichnen. Komfortabel wäre eine grafische Ausgabe als Scalable Vector Graphic (SVG). Recherche bei w3c.org und Adobe.
- Halten Sie sich beim Entwurf und bei der Implementierung strikt an das Divide-and-Conquer-Paradigma!
- Jeder Knoten und jede Kante des Baumes muss als GUI-Objekt repräsentiert werden und mit der Maus aktivierbar sein. Die Attribute der GUI-Objekte sollen bei Aktivierung angezeigt werden.
- Definieren Sie den Baum als XML-Struktur und lesen Sie diese ein. Auch hierfür gibt es bereits Java-Klassen.
- Fügen Sie interaktiv Knoten zum Baum hinzu und rufen Sie danach den *Layered-Tree-Draw-Algorithmus* auf, um den Baum neu zu zeichnen.
- Die GRID-Größe soll über einen Parameter modifizierbar sein. Entsprechend müssen die Knotengrößen beim Zeichnen angepasst werden.
- Erzeugen Sie aus Ihren Java-Klassen ein jar-File und legen Sie dieses auf Ihrem ADV-Account ab.

- Fertigen Sie eine ausführliche Dokumentation an. Erklären Sie, welche Eigenschaften die Zeichnung Γ des Baumes hat.

Links:

Java-Bibliothek: <http://www.cs.hut.fi/Research/Matrix/>

SVG-Spezifikation: <http://www.w3.org/TR/SVG/>

SVG-PlugIn <http://www.adobe.com/svg/>

SVG-Tool <http://xml.apache.org/batik/>

<http://www.sun.com/software/xml/developers/svg/>