

AD Skizze Aufgabe 4

Team: <3>, <Patrick Steinhauer, Jan Dennis Bartels>

Aufgabenaufteilung:

Die Aufgaben Wurden zusammen bearbeitet und erstellt. Jeder hat sich was für die Skizze Überlegt, und dann wurde es zusammengefügt.

Quellenangaben: VorlesungsFolien Teilweise Wikipedia

Begründung für Codeübernahme:

Bearbeitungszeitraum: Ca 5 Stunde Bisher

Aktueller Stand: Skizze + Grundgerüst der AVL Bäume

Änderungen in der Skizze:

Skizze:

Generelles:

Balance :

- Die Balance muss nach jedem einfügen, löschen, rotieren neu berechnet werden.
- Die Balance errechnet sich aus :
 - Höhe rechter Teilbaum – Höhe linker Teilbaum
 - Bei einem Blatt zieht man die Höhe von sich selbst ab, also 1-1
 - Die Wurzel immer mitzählen

Signatur der Methoden:

Create: $\{\}$ \rightarrow AVL Baum

Add : AVL Baum x elem \rightarrow AVL Baum

Delete: AVL Baum x elem \rightarrow AVL Baum

rotateRight: AVL Baum \rightarrow AVL Baum

rotateLeft: AVL Baum \rightarrow AVL Baum

rotateLeftRight: AVL Baum \rightarrow AVL Baum

rotateRightLeft: AVL Baum \rightarrow AVL Baum

Create :

- Das Create soll einen leeren AVL Baum zurückgeben, der in unserem Fall so aussieht $\rightarrow \{\}$

Add :

- Das Add soll in einen AVL Baum ein bestimmtes Element nach Definition einsortieren.

Delete :

- Das Delete soll aus einem bestehenden AVL Baum ein bestimmtes Element herauslöschen.

rotateRight :

- Das rotateRight rotiert den AVL Baum, falls die Balance des AVL Baumes verletzt wurde. Dies geschieht, wenn der obere Knoten -2 als Wert hat und der darunterliegende -1. Wie der Name schon sagt wird nach rechts rotiert.

rotateLeft :

- Das rotateLeft rotiert den AVL Baum, falls die Balance des AVL Baumes verletzt wurde. Dies geschieht, wenn der obere Knoten +2 als Wert hat und der darunterliegende +1. Wie der Name schon sagt wird nach links rotiert.

rotateLeftRight :

- Das rotateLeftRight rotiert den AVL Baum, falls die Balance des AVL Baumes verletzt wurde. Dies geschieht, wenn der obere Knoten -2 als Wert hat und der darunterliegende +1. Wie der Name schon sagt wird erst nach links rotiert, danach nach rechts.

rotateRightLeft :

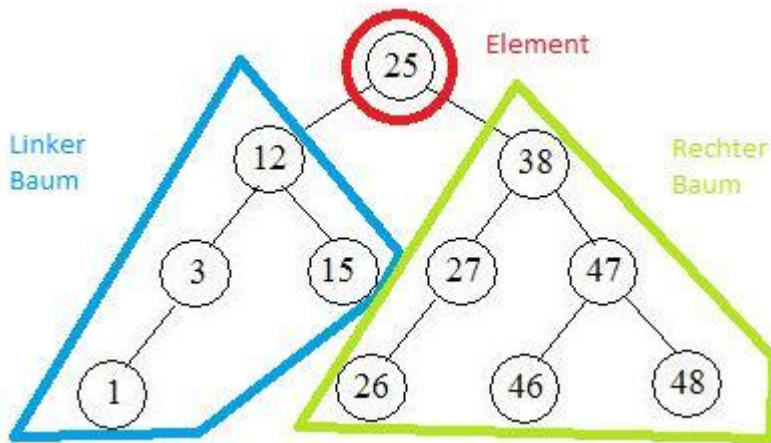
- Das rotateRightLeft rotiert den AVL Baum, falls die Balance des AVL Baumes verletzt wurde. Dies geschieht, wenn der obere Knoten +2 als Wert hat und der darunterliegende -1. Wie der Name schon sagt wird erst nach rechts rotiert, danach nach links.

Pseuso Codes:

Aufbau eines AVL Baumes: {LinkerUnterBaum, {Element, Höhe}, RechterUnterbaum}

Eine weitere mögliche Variante sieht so aus:

{LinkerUnterBaum, Element, RechterUnterbaum}



einfügen / add

```
add(Baum, Value)
{
    Wenn (Baum == empty){
        Baum = Value
    }
    Wenn (Baum.Elem < Value && Baum.Rechts == empty) {
        Baum.Rechts = Value, rotate(Baum)
    }
    Wenn(Baum.Elem < Value && Baum.Rechts != empty) {
        add(Baum.Rechts, Value)
    }
    Wenn (Baum.Elem > Value && Baum.Links == empty) {
        Baum.Links = Value, rotate(Baum)
    }
    Wenn(Baum.Elem > Value && Baum.Links != empty) {
        add(Baum.Links, Value)
    }
}
```

Löschen :

Generelles:

- Nach dem eine Element gelöscht wurde muss die höhe des Baumes neu berechnet werden.

```
delete(Baum,Elem) {
  if(Baum.Value == Elem) {
    if (Baum.Right == empty) {
      Baum.Value = Baum.Left, Baum.Left = NULL;
    }
    else if (Baum.left == empty) {
      Baum.Value = Baum.right, Baum.right = NULL;
    } else {
      newElem = findSmallest(Baum.right);
      Baum.Value = newElem;
      Baum.right = delete(Baum.right,newElem);
      rotate(Baum);
    }
  } else if (Baum.Value > ELEM) {
    delete(Baum.Right,Elem);
    rotate(Baum);
  } else {
    delete(Baum.Left,Elem);
    rotate(Baum);
  }
}
```

Für alle rotate Funktionen eine Funktion machen die eine Unterscheidung für alle Fälle macht.

RechtsRotation :

Generelles :

- Eine Rechtsrotation wird durchgeführt, wenn
 - Die Balance des oberen Knotens -2 ist
 - Die Balance des unteren Knotens -1 ist

Wenn also Baum.Element == -2 und unterbaum.Element == -1 dann

```
rotateRight(Baum) {  
}
```

LinksRotation :

Generelles :

- Eine Linksrotation wird durchgeführt, wenn
 - Die Balance des oberen Knotens +2 ist
 - Die Balance des unteren Knotens + 1 ist

```
rotateLeft(Baum) {  
  
}
```

LinksRechtsRotation (Doppelrotation rechts) :

Generelles:

- Eine Linksrechtsrotation wird durchgeführt, wenn
 - Die Balance des oberen Knotens -2 ist
 - Die Balance des unteren Knotens +1 ist

```
rotateLeftRight(Baum) {  
  
}
```

RechtsLinksRotation (Doppelrotation Links)

Generelles :

- Eine RechtsLinksRotation wird durchgeführt, wenn
 - Die Balance des oberen Knotens +2 ist
 - Die Balance des unteren Knotens -1 ist

```
rotateRightLeft(Baum) {  
  
}
```