Dubbelrotationer i AVL-träd:

Ett AVL-träd är ett binärt träd där skillnaden i djup mellan de olika grenarna i trädet ej får vara större än 1. Då skillnaden blir för stor, behöver en rotation utföras. Detta är för att den logaritmiska sökalgoritmen ska kunna behållas i träden, då den närmar sig linjär om trädet blir för obalanserat. Det gör att det tar längre tid att söka och lägga till element i trädet.

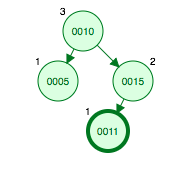
En enkelrotation är en rotation som sker i ett träd som blir obalanserat, om det lövet som ger obalansen finns i en rak linje från dess förförälder.

I vissa fall kan en enkelrotation inte lösa problemet med obalansen i trädet, och en dubbelrotation måste utföras. I regel behöver dubbelrotation göras då ett zigzagmönster uppstår mellan förföräldernoden och det löv som gav obalansen.

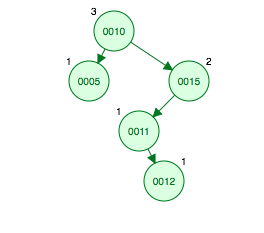
En dubbelrotation går till så att den lövnod som skapade obalansen blir förälder till den nod som tidigare var förälder, samt till den nod som tidigare var barn. Om denna rotation ej löser obalansen i trädet, utförs ej rotationen. Istället testat rotationen att utföras med noden som är förälder till förföräldernoden till lövet. Detta testas om och om igen tills en rotation sker som gör att obalansen i trädet försvinner.

Exempel:

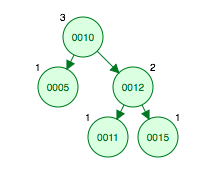
Noden 12 ska sättas in i trädet.



Insättningen ger en obalans i trädet, och grenen där lövet sätts in blir för tung.



En dubbelrotation utförs, eftersom det blir ett zigzag-mönster mellan lövet och de två noder som finns ovanför den. Den nya lövnoden blir förälder, medan den tidigare föräldranoden blir barn på höger sida, och den tidigare barnnoden blir barn på vänster sida.



Regler:

1: Se om obalans finns.

2: Se om obalansen ger ett zigzag-mönster 🡪Dubbelrotation eller en rak linje 🡪enkelrotation

3: Se om en rotation skulle lösa problemet

4: Om inte, gå upp en nod

5: Repetera steg 2-4 tills balans är funnen