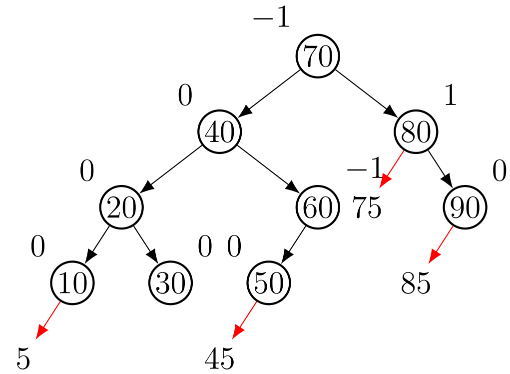
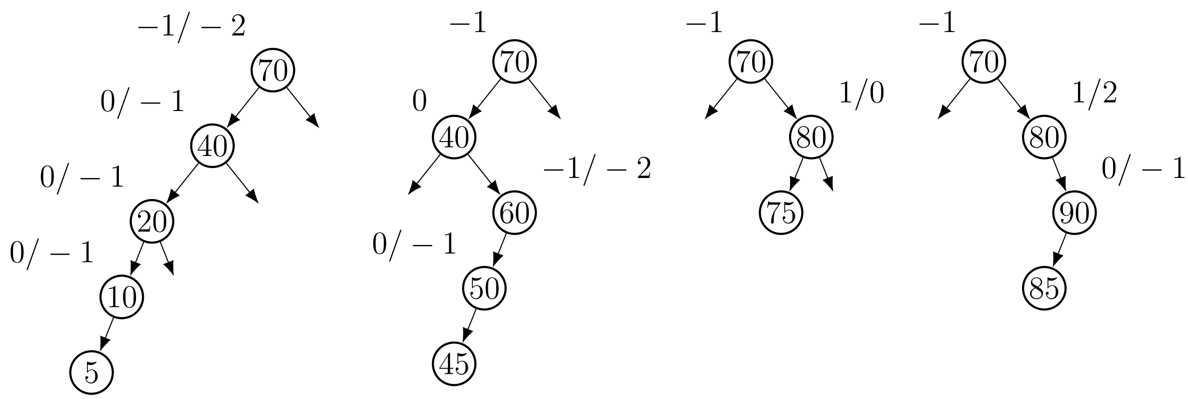
2. **a)** Toevoegen bij een AVL boom gebeurt op de plek zoals bij een gewone binaire zoekboom. Hieronder staan die posities aangegeven voor de vier waardes. Ook staan de balansfoctoren gegeven zoals voor de oorspronkelijke boom, dus vóór toevoegen.



Bij toevoegen verandert de balansfactor van de knopen langs het pad waar toegevoegd is. Dit doen we gewoonlijk van onder naar boven. Bij $0/\pm 1$ vervolgen we naar boven. Bij $\pm 1/0$ kunnen we stoppen, de boom onder de huidige knoop is niet dieper geworden. Bij $\pm 1/\pm 2$ is de knoop uit balans. Op dat punt kan de boom hersteld worden met een (enkele of dubbele) rotatie.



(5) Onbalans bij 70, de wortel. Situatie LL, enkele rotatie naar rechts.

(45) Onbalans bij 60. Weer situatie LL, enkele rotatie naar rechts. (De rotatie zorgt ervoor dat de hoogte bij 60 gelijk is die in de oorspronkelijke situatie. We hoeven hogerop geen balansfactoren aan te passen.)

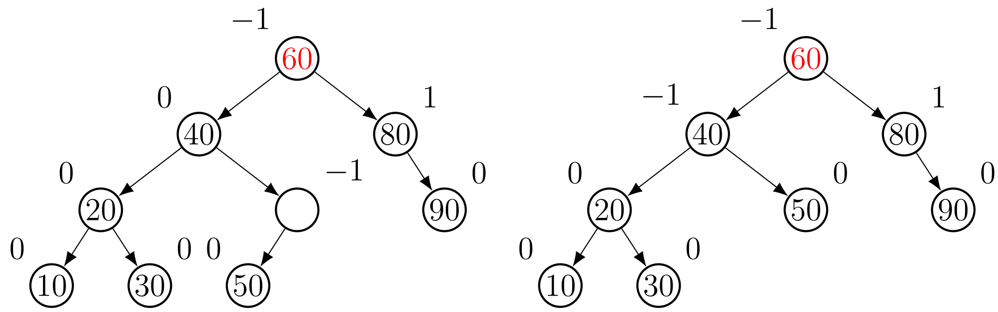
(75) Bij 80 onstaat balans 0. Klaar, geen rotatie.

(85) Onbalans bij 80. Situatie RL, dubbele rotatie naar links.

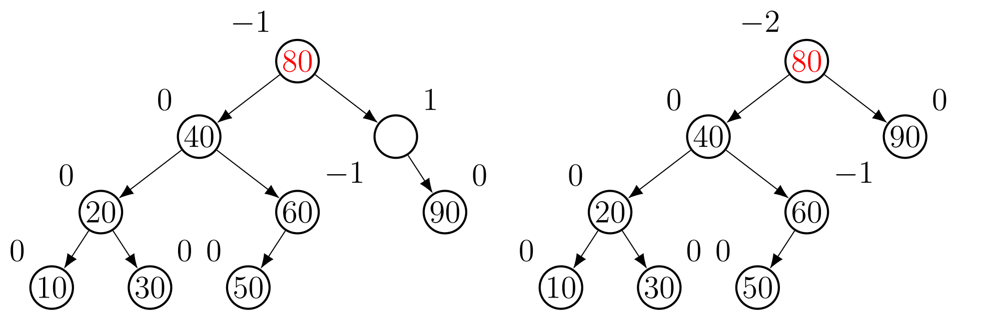
b) Verwijder de waarde 70 uit boom *A*. Dit kan op twee manieren, laat deze beide zien, met resultaat.

De knoop die we moeten verwijderen heeft twee kinderen. We passen *deletion by copying* toe. Verplaats (kopieer) de voorganger (of de opvolger) van 70 naar die plek.

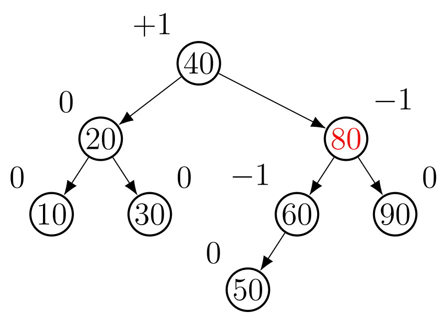
Voorganger. Verplaats 60 naar 70. De oorspronkelijke knoop 60 heeft één kind 50. Hang de subboom 50 op de plaats van de oude 60, dus als rechterkind van 40. Er ontstaat (toevallig) een AVL-boom.



Opvolger. Verplaats 80 naar 70. De oorspronkelijke knoop 80 heeft één kind 90. Hang de subboom 90 op de plaats van de oude 80, dus als rechterkind van de nieuwe 80. De nieuwe boom heeft onbalans in de wortel. De boom is dieper naar links, situatie LL, dus we roteren enkel naar rechts. (* Zie beneden)



Het resultaat is gelukkig een AVL-boom. (In het algemeen kunnen er meerdere rotaties mogelijk zijn bij verwijderen in AVL-bomen.)



Opmerking. (*) We zouden ook situatie LR kunnen nemen en dubbel roteren. Dat gaat hier ook goed, maar dat is niet algemeen geldig. Als we kunnen kiezen dan liever enkel roteren.

c) **(i)** Zoals in de opgave tellen we hier hoogte in het aantal knopen. (De opgave is eenvoudiger dan ik oorspronkelijk dacht. Er is geen inductie nodig. . .)

Als een knoop in een binaire boom hoogte h heeft, dan heeft tenminste een van de kinderen hoogte $h-1$, en het andere andere kind maximaal hoogte $h-1$. Als de boom ook een AVL-boom is kan de hoogte van de kinderen maximaal 1 één verschillen. Dus het andere kind heeft hoogte $h-1$ of hoogte $h-2$.

In het slechtste geval verschilt er steeds een kind twee in hoogte van de ouder. Als we starten met hoogte $2k$ of $2k-1$ zijn er in die reeks k knopen voordat we hoogte nul kunnen krijgen. Dat betekent dat er k nivo's gevuld zijn.

Voor het slechtste geval kun je kijken naar de Fibonacci-boom in het diktaat.

(ii) Toevoegen van een waarde in een AVL-boom loopt langs een pad in de boom vanaf de wortel om de plek te vinden, en dan weer terug om de balansfactoren aan te passen en eventueel te roteren op een plek met disbalans. Bij verwijderen geldt dat ook, behalve dat we eventueel op elk nivo moeten roteren. De orde-complexiteit van een operatie is dus lineair in de hoogte van de boom.

Als de boom hoogte h heeft dan zijn er tenminste $h/2$ nivo's geheel gevuld, zie hierboven. Dat betekent dat er tenminste $n = 2^{h/2} - 1$ knopen zullen zijn, exponentieel in de hoogte. Omgekeerd, als we de complexiteit zoals gewoonlijk uitdrukken in het aantal knopen zien we dat de hoogte logaritmisch is (in het aantal knopen).