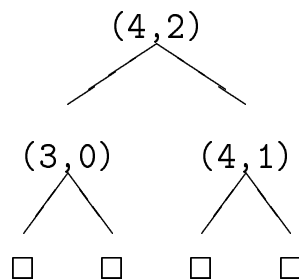


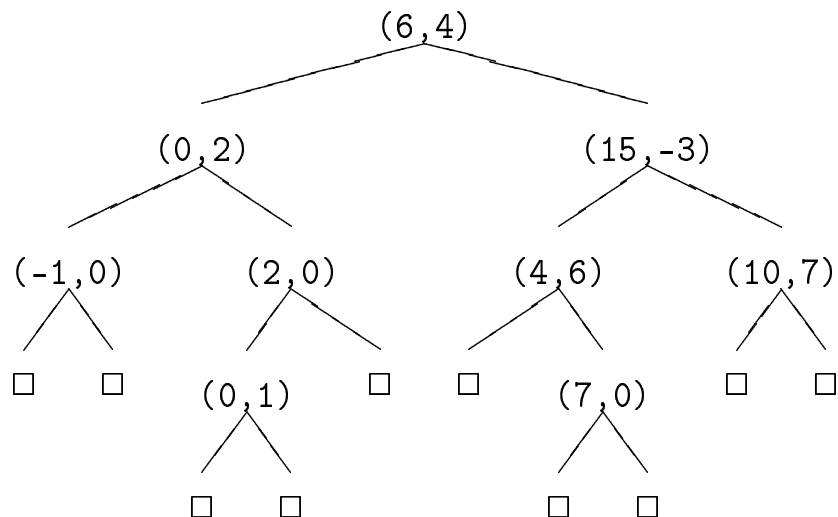
Opgave 11

I denne opgave ser vi på en variant af *søgetræer*, der indeholder heltal. Den sædvanlige implementation er modificeret således, at hver knude foruden en værdi indeholder en *addend*, der implicit er lagt til alle værdier i det undertræ, som den er rod i. Indholdet af en knude angives som et par $(værdi, addend)$. For eksempel indeholder træet



elementerne 5, 6 og 7.

a) Hvilke elementer indeholder søgetræet



b) Angiv, hvordan de følgende operationer kan realiseres, hvor vores nye datastruktur kaldes `VTree`. `vt` er en variabel af type `VTree`

- `new VTree()` i tid $O(1)$

- **vt.member**(i) og **vt.insert**(i) i tid $O(h)$
- **vt.plus**(k) i tid $O(1)$. Denne operation lægger tallet k til alle elementer i vt .
- **vt.skub**(i, k) i tid $O(h)$. Denne operation lægger tallet $k \geq 0$ til alle de elementer i vt , der er $\geq i$.

Her angiver h højden af det aktuelle søgetræ.

- c) Vis, hvordan man kan sikre, at højden af søgetræerne altid er logaritmisk i antallet af elementer i søgetræet. Dette kan man fx gøre ved at vise hvordan den rød-sort strategi kan tilpasses denne variant.