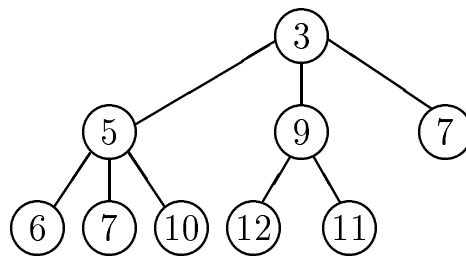


Opgave

En heap er iflg. [G&T] afsnit 6.3 et perfekt balanceret *binært* træ, hvor det nederste lag er fyldt så meget som muligt fra venstre mod højre. En *d*-heap er ligeledes et perfekt balanceret træ (igen eventuelt bortset fra nederste lag), men træets indre knuder har nu *d* efterfølgere i stedet for blot 2 – en heap som beskrevet i [G&T] er således en 2-heap. Følgende er et eksempel på en 3-heap.



- Vis, hvorledes en prioritetskø kan implementeres som en *d*-heap (for et vilkårligt $d \geq 2$), og angiv kompleksiteten af prioritetskøens operationer.
- Antag, at en *d*-heap indeholder n elementer. Angiv kompleksiteten af indsættelse og fjernelse når
 - $d = 3$
 - $d = \log n$
 - $d = \sqrt{n}$
 - $d = n$
- Efter hvilke kriterier skal man vælge d i en given anvendelse?