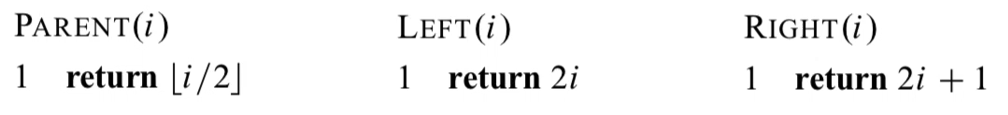
**Rooted trees**

Arrays kan repræsenteres som binære træer hvor man deler index ud i træet hvor index 1 er roden af træet. Binære træer kan have et LEFT og RIGHT barn og et parent. De 3 Algoritmer under viser hvordan man ”bevæger” sig i træet.



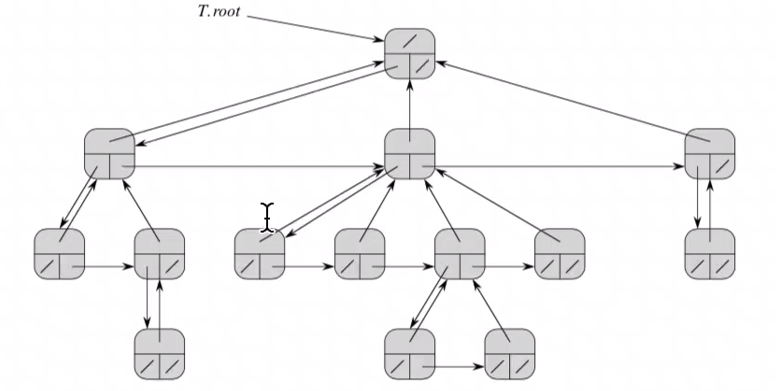
Nogle træer kan ikke repræsenteres sådan. Det kræver indexes er opstillet korrekt. F.eks. Hvis en node i træet har et RIGHT child, men intet LEFT child kan det ikke repræsenteres (pga. måden indexes virker).

**Binære træer**

Repræsentationen af træer bruger pointers. Et træ T har attributen **T.root**, som er en pointer til roden af træet. **X.p** er en pointer til parent noden (hvis x.p er NIL er det roden). Til sidst har træer også **x.left** og **x.right** som er pointers til børnene (hvis de pointer til NIL er der ingen børn). Det kan også være en node x bruges til at holde data.

**Rooted træer med mere end 2 børn**

Rooted trees kan bruges til at repræsentere mere end binære træer. Her bruges der Left-child right sibling. T.root og x.p er stadig det samme som de binære træer. Dog hvis en node har mere end 3 børn kan den ikke repræsenteres som et binært træ. Her bruges der stadig **x.left-child**, men i stedet for x.right bruges der **x.right sibling. X.right-sibling** er den node til højre for x. Eksempel:



Dette kan ses som en singly-linked list for børnene.

A screenshot of a cell phone

Description automatically generated