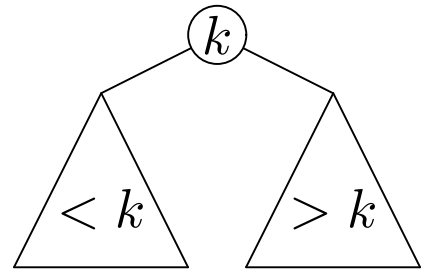


# Dictionary (ordbog)

insert, delete, sgning

## Binære søgetræer

Nøglen i en knude er større end nøglerne i venstre undertræ og mindre end nøglerne i højre undertræ.



insert( $x$ ): Find vha. search den plads i bunden af træet, hvor  $x$  passer ind, og indsæt den her.

delete( $x$ ): Find knuden med  $x$  vha. search.

Knuden med  $x$  har 0, 1 eller 2 børn:

0 : knuden med  $x$  kan blot slettes.

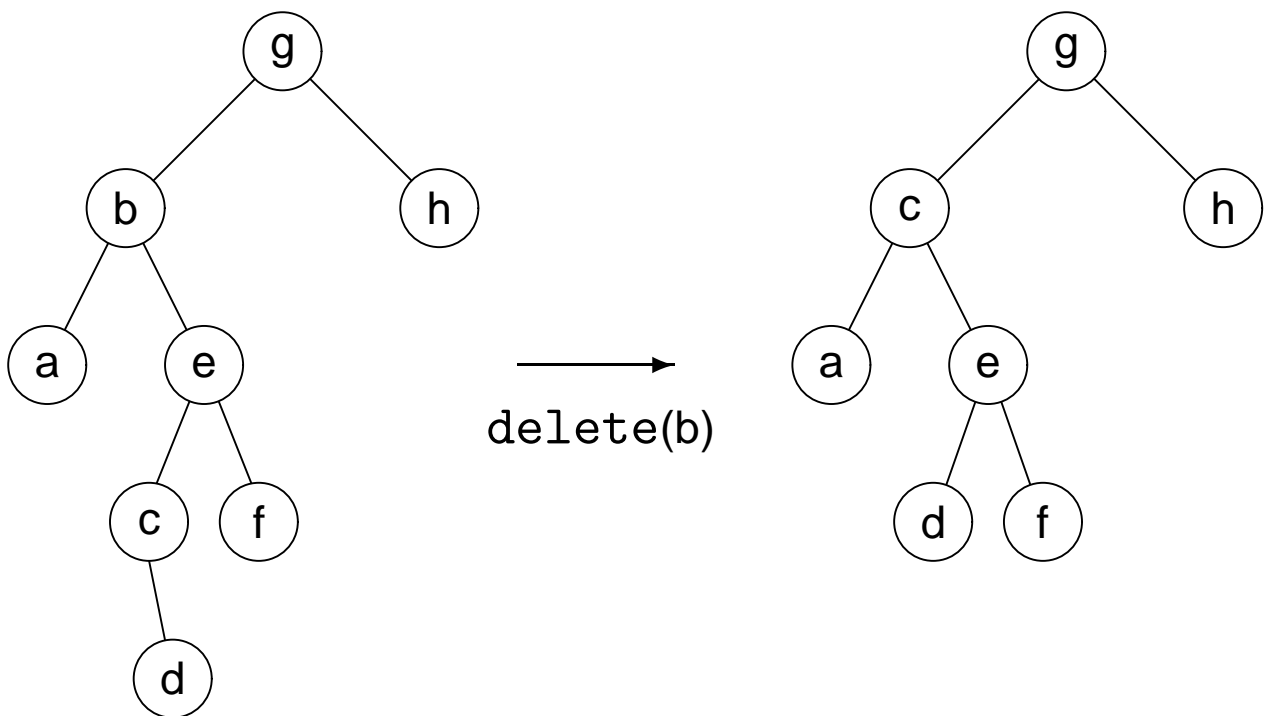
1 :  $x$  overskrives med elementet i barnet, og barnet slettes.

2 :  $x$  overskrives med sin efterfølger  $y$ , som er længst til venstre i  $x$ 's højre undertræ. Derefter slettes knuden, som indeholdt  $y$  (den har 0 eller 1 barn).

Logisk slettede knude: Knuden hvis indhold forsvinder.

Strukturelt slettede knude: Knuden som forsvinder.

**Eks:**



Logisk slettede knude: b

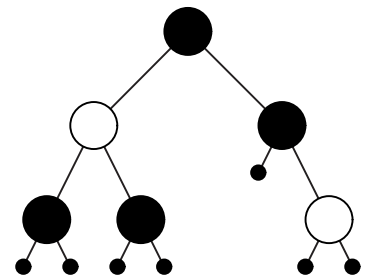
Strukturelt slettede knude: c

## Rød-sorter søgetræer

insert, delete, search:  $\Theta(\log n)$

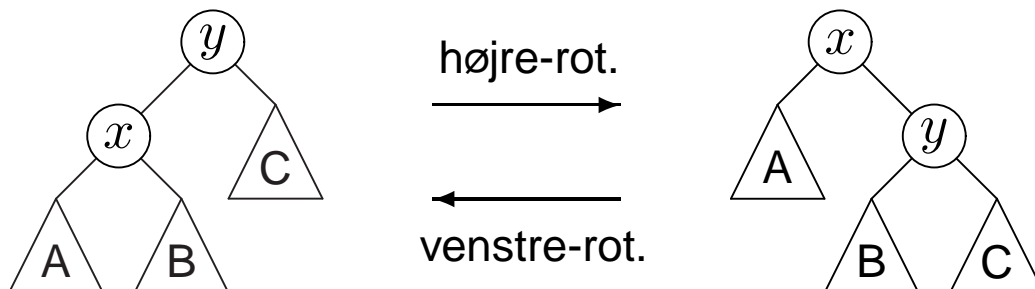
Logaritmisk højde sikres ved at opretholde flg.

- For enhver knude  $v$  gælder:  
alle stier fra  $v$  til et blad har  
samme # sorte knuder
- Ingen rød knude har et rødt barn.
- Roden og bladene er sorte.



Til dette formål anvendes rotationer og farveskift.

insert og delete udføres som i alm. binære søgetræer, efterfulgt af  $O(1)$  rotationer og  $O(\log n)$  farveskift.



insert :

Den indsatte knude farves rød.

Hvis forælderen er

- rød: ryd op!
- sort: færdig!

delete :

Hvis den strukturelt slettede knude er

- rød: færdig!
- sort: ryd op!