

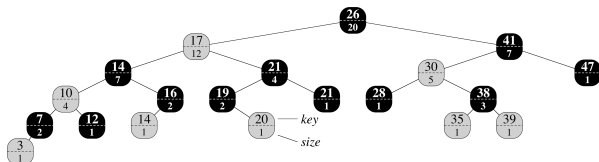
Binære søgetræer med ekstra information i
knuderne

Tilføj ekstra information i knuderne

Konkret eksempel på ekstra information i knuder:

Alle knuder gemmer størrelsen af deres undertræ (dvs. antallet af knuder i deres undertræ).

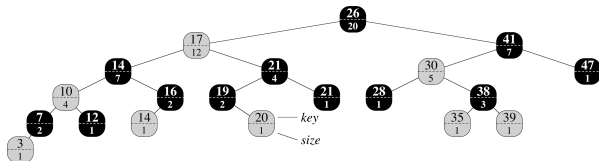
Et binært søgetræ (rød-sort) med denne information tilføjet:



Ny funktionalitet

Målet med ekstra information i knuderne er at tilføje ekstra funktionalitet. F.eks. kan man i eksemplet ovenfor (med størrelse af undertræer gemt i knuder) udføre flg. operationer i $O(\log n)$ tid:

- Find rang.
- Find nøgle som har en given rang.



Her er **rang** elementets nummer i sorteret orden (blandt de gemte).

Vedligehold den ekstra information i knuderne

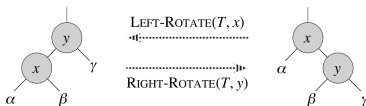
Antag følgende egenskab gælder for den ekstra information:

Hvis en knudes to børns værdier k_1 og k_2 allerede er korrekte, kan knudes egen værdi k beregnes i $O(1)$ tid. Blades værdier kan beregnes i $O(1)$ tid.

I eksemplet ovenfor: k kan findes som $1 + k_1 + k_2$.

Af antagelsen følger at værdierne kan vedligeholdes under updates uden ændring af $O(\log n)$ køretid:

- ▶ Indsættelse af ny knude: opdatér nedefra og op langs sti fra ny knude til roden (ingen andre knuder skal have værdi ændret).
- ▶ Sletning af knude: samme.
- ▶ Under den efterfølgende rebalancering: genberegner for berørte knuder efter hver rotation og efter hvert skridt opad.



Andre eksempler

Hvis en knudes to børns værdier k_1 og k_2 allerede er korrekte, kan knudes egen værdi k beregnes i $O(1)$ tid.

Eksempler på anden information som kan vedligeholdes på samme måde:

- ▶ Max af data-værdier i undertræet.
- ▶ Min af data-værdier i undertræet.
- ▶ Sum af data-værdier i undertræet.

Sådanne data kan tilføje yderligere funktionalitet til træet.