Binære søgetræer, herunder rød/sorte træer

Mads Ωhm Larsen

6. april 2008

1 Binære søgetræer

1.1 Definition

Binært træ, altså barn til venstre for en knude **skal** være mindre, og barn til højre **skal** være større. Hver knude har en nøgle og højre, venstre og forfæder pegere.

1.2 Operationer

HUSK: højden kan være lig med antallet af elementer!

1.2.1 Max og min

Går helt til højre eller helt til venstre alt efter hvad man skal bruge. Tager $O(h)$ hvor h er højden af træet.

1.2.2 Direkte forgænger og efterkommer

Finde den største eller mindste værdi der kommer lige efter denne. Tager $O(h)$ tid.

1.2.3 Indsæt

Går helt ned i bunden og sætter den ind hvor den passer. Tager $O(h)$ tid.

1.2.4 Slet

Der er 3 tilfælde når vi skal slette en knude z

1. Ingen børn. Trivielt. Bare slet knuden z
2. 1 barn. Sæt z's parent som parent på z's barn.
3. 2 børn. Find direkte efterkommer. Har denne to børn, prøv igen. Fjern den direkte efterkommer (enten ved (1) eller (2)), og byt den ud med z.

2 Rød/sort træer

Vi vil finde en bedre løsning. En hvor vi kan begrænse højden, da denne spiller en stor rolle for de mest brugte operationer. Vi vil balancere træet.

2.1 Definition

Binært søgetræ, men med 5 specielle egenskaber:

1. Alle knuder er enten røde eller sorte
2. Roden er altid sort
3. Alle blade er sorte (`NIL`)
4. Hvis en knude er rød er dens to børn sorte
5. Alle veje fra en knude til et blad skal igennem det samme antal sorte knuder

2.2 Bevis for øvre grænse for sort højde

Vi har nu lavet disse definitioner, men skal så bevise at de holder. Vi vil bevise at højden for et rød/sort træ højst kan være $2\lg(n+1)$ for n indre knuder.

2.2.1 Bevis for antal af indre knuder

Vi vil starte med at vise at et undertræ, x, har mindst $s(x) \geq 2^{bh(x)} - 1$ indre knuder. Dette kan vises ved induktion på højden (HUSK: Ikke sort højden, men højden på træet!).

Hvis højden er 0, må undertræet, x, være et blad og i følge vores antagelse vil det indeholde

$$s(0) \geq 2^{bh(0)} - 1 = 2^0 - 1 = 0$$

$bh(0)$ er 0, da det er et blad, og har altså ingen børn.

Vi antager at dette holder for alle $x < k$. Lader vi nu $x = k$, vil antallet af indre knuder være

$$s(x) \geq s(V(x)) + s(H(x)) + 1$$

Antagelsen må holde for de to undertræer $V(x)$ og $H(x)$. Skifter vi de to børn ud med antagelsen, og samtidig bemærker at afhængig af om knuden er sort eller rød har den sort højden $bh(x)$ eller $bh(x) - 1$, har vi

$$s(x) \geq 2^{bh(x)-1} - 1 + 2^{bh(x)-1} - 1 + 1 = 2^{bh(x)} - 1$$

Lader vi højden være h og samtidig bemærker at eftersom begge børn for en rød knude er sort, må sort højden være mindst $h/2$ altså:

$$n \geq 2^{h/2} - 1$$

Vi kan nu vise at højden er begrænset til

$$\begin{aligned} n + 1 &\geq 2^{h/2} \\ \lg(n+1) &\geq \frac{h}{2} \\ 2\lg(n+1) &\geq h \end{aligned}$$

2.3 Reetablering af træet

Ved indsættelse og sletning kan vi komme til at overskride vores definition for rød/sorte træer. Vi skal derfor reetablerer træet. Dette kan gøres ved at sige at alle knuder der bliver indsat er røde, ellers ville det lave vrøvl med sort højden igennem hele træet. Vi kan også bruge rotationer til at reetablerer træet, her propagere vi problemet op gennem træet.