

Ugeseddel: Prioritetskøer og hobe

Philip Bille

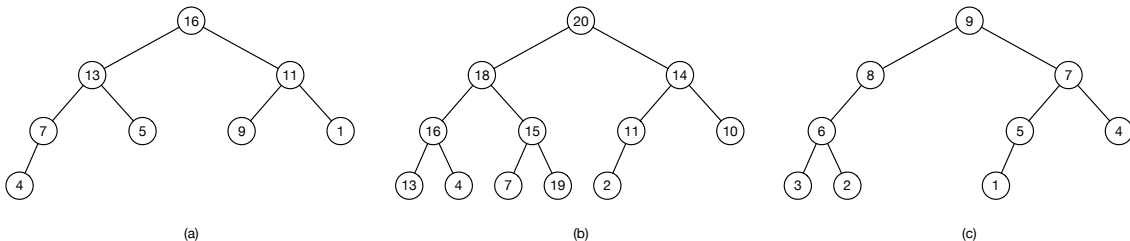
Om denne uge

Litteratur *Introduction to Algorithms*, Cormen, Rivest, Leisersons og Stein (CLRS): Kap. 6 + appendix B.5

Opgaver

1 Hobegenskaber og håndkøring Løs følgende opgaver.

1.1 [o] Hvilke af følgende træer er en hob?



1.2 [o] Hvilke af følgende tabeller er en hob? Index 0 som ikke bruges er markeret med –

$$A = [-, 9, 7, 8, 3, 4] \quad B = [-, 12, 4, 7, 1, 2, 10] \quad C = [-, 5, 7, 8, 3]$$

1.3 [o] Lad $S = 4, 8, 11, 5, 21, *, 2, *$ være en sekvens af operationer hvor alle tal svarer til en indsættelse af tallet og $*$ svarer til en EXTRACTMAX operation. Startende med en tom hob H , vis hvordan H ser ud efter hver operation i S .

1.4 Er en sorteret tabel en hob?

1.5 Hvor kan man finde det mindste element i en (max-)hob?

1.6 [C] Vis at INSERT, EXTRACTMAX og INCREASEKEY bibeholder hoborden.

1.7 [*] CLRS 6.5-9.

2 [o] Prioritetspolitik Teknokratisk alternativ vil gerne have hjælp til at implementere deres "frisk luft"-politik. Der skal designes et register over borgere og deres indkomst, således man effektivt kan finde dem lavest indkomst og deportere dem. Specifik skal systemet understøtte følgende operationer.

- $INDSÆT(c, i)$: indsæt person med cpr. nr. c og årlig indkomst i i systemet.
- $SLETLAVESTEINDKOMST()$: fjern og returner (deporter) person med laveste indkomst.

Foreslå en effektiv datastruktur til systemet.

3 Prioritetskøoperationer Vi vil gerne tilføje nogle operationer til vores prioritetskø. Vi er interesseret i at tilføje følgende operationer.

- REMOVELARGEST(m): fjern de m største elementer i hoben.
- DELETE(x): fjern elementet x fra hoben.
- FUSION(x, y): fjern x og y fra hoben og tilføj elementet z med nøgle $x.key + y.key$.
- FINDLARGEST(x): returner de elementer i hoben med nøgle $\geq x$.
- EXTRACTMIN: fjern og returner element med mindste nøgle.

Vi vil gerne implementere disse operationer, mens vi stadig bibeholder kompleksiteten af de sædvanlige prioritetskøoperationer. Lad n være antallet af elementer i prioritetskøen. Løs følgende opgaver.

- 3.1 Udvid prioritetskøen til at understøtte REMOVELARGEST(m) i $O(m \log n)$ tid.
- 3.2 Udvid prioritetskøen til at understøtte DELETE og FUSION i $O(\log n)$ tid.
- 3.3 [*] Udvid prioritetskøen til at understøtte FINDLARGEST i $O(m)$ tid, hvor m er antallet af elementer med nøgle $\geq x$.
- 3.4 [*] Udvid prioritetskøen til også at understøtte EXTRACTMIN i $O(\log n)$ tid.

4 Satellitdata Lad $A[0..n]$ være en tabelrepræsentation af en hob. Hvert element x i hob er repræsenteret ved et indeks i og nøglen gemt i $A[i]$. Vi er ofte interesseret at gemme noget ekstra information (kaldet *satellitdata*) associeret med elementet (f. eks. hvis vi gerne vil gemme personer i en hob kunne satellitdata være alder, køn, højde, vægt, etc.). Vis hvordan man kan understøtte tilgang til satellitdata i $O(1)$ tid kun givet indeks i , mens man stadig bibeholder køretiden for de normale hoboperationer.

5 Hobegenskaber Lad T være et komplet binært træ af højde h . Løs følgende opgaver.

- 5.1 Vis at antallet af knuder i T er $n = 2^{h+1} - 1$. *Hint*: vi har at $n = 1 + 2 + 4 + \dots + 2^h$. Multipliser summen med 2 og træk summen fra.
- 5.2 [C] Vis at summen, $S = n/4 \cdot 1 + n/8 \cdot 2 + n/16 \cdot 3 + n/32 \cdot 4 + \dots = \Theta(n)$. *Hint*: Udregn $S - S/2$.

6 Implementation af hobe Vi er interesseret i at implementere en prioritetskø vha. en hob med tabelrepræsentationen. Løs følgende opgaver.

- 6.1 [†] Implementer INSERT og EXTRACTMAX operationen.

7 Delsummer Lad $A[0..n-1]$ være en tabel af heltal. Vi er interesseret i følgende operationer på A .

- SUM(i, j): beregn $A[i] + A[i+1] + \dots + A[j]$.
- CHANGE(i, x): sæt $A[i] = x$.

Løs følgende opgaver.

- 7.1 [o] Giv en datastruktur, der understøtter SUM i $O(1)$ tid og bruger $O(n^2)$ plads.
- 7.2 [*] Giv en datastruktur, der understøtter SUM i $O(1)$ tid og bruger $O(n)$ plads.
- 7.3 [**] Giv en datastruktur, der understøtter både SUM og CHANGE i $O(\log n)$ tid og bruger $O(n)$ plads.

O Obligatorisk afleveringsopgave: Hobe og tabeller Løs følgende opgaver.

- O.1 Lad A være en tabel af længde n , $n \geq 1$. Giv en algoritme, der afgør om A er en hob.
- O.2 Implementer din algoritme og beskriv kort din implementation.
- O.3 Skriv et program, der afprøver din implementation på et par eksempler. Afprøv dit program og kommenter på resultaterne.