

Prioritetskøer og hobe

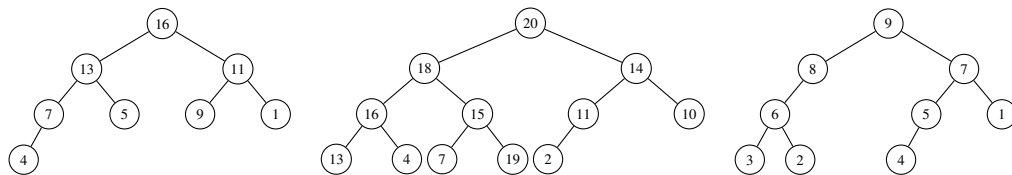
Eva Rotenberg*

Materialer CLRS: kap. 6 + appendix B.5.

Opgaver

1 Hobegenskaber og håndkøring

1.1 Hvilke af følgende træer er en hob?



1.2 Hvilke af følgende arrays er en hob? Indeks 0 som ikke bruges er markeret med $-$:

$$A = [-, 9, 7, 8, 3, 4]$$

$$B = [-, 12, 4, 7, 1, 2, 10]$$

$$C = [-, 5, 7, 8, 3]$$

1.3 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 et sorteret array en hob?

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

1.6 [*] Givet k sorterede arrays med i alt n elementer, vis, at disse kan flettes til ét sorteret array i $O(n \log k)$ tid.

2 Stubbekøbing Universitet

Regeringen vil gerne flytte uddannelser fra de etablerede universiteter til det nye universitet i Stubbekøbing. De har ansat nogle dygtige konsulenter til at vurdere alle uddannelsernes kvalitet, og ønsker at kunne flytte uddannelserne af lavest kvalitet, i håb om at den friske havluft helt automatisk forbedrer uddannelsen. Specifikt ønsker de et system, der kan håndtere følgende operationer:

- `INDSÆT(u, k)`: indsæt uddannelse nr. u med konsulentvurdering k i systemet.
- `UDFLYTRINGESTEUDDANNELSE()`: fjern og returnér uddannelse med laveste vurdering.

Foreslå en effektiv datastruktur til systemet.

*baseret på materiale af Billes&Gørtz

3 Prioritetskøoperationer

Vi vil gerne tilføje nogle operationer til vores prioritetskø. Vi antager en max-hob i arrayrepræsentation der som udgangspunkt kun understøtter de fire kendte operationer $\text{MAX}()$, $\text{EXTRACTMAX}()$, $\text{INCREASEKEY}(x, k)$ og $\text{INSERT}(x)$. Vi er interesseret i at tilføje følgende operationer:

- $\text{REMOVELARGEST}(m)$: fjern de m største elementer i hoben.
- $\text{DELETE}(x)$: fjern elementet x fra hoben.
- $\text{FUSION}(x, y)$: fjern x og y fra hoben og tilføj elementet z med nøgle $x.\text{key} + y.\text{key}$.
- $\text{FINDLARGEST}(x)$: returnér de elementer i hoben med nøgle $\geq x.\text{key}$.
- EXTRACTMIN : fjern og returnér 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. Du må kun bruge ekstra plads $O(n)$. Løs følgende opgaver:

- 3.1 Udvid prioritetskøen til at understøtte $\text{REMOVELARGEST}(m)$ 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 arrayrepræsentation af en hob. Hvert element x i hoben er repræsenteret ved et indeks i og nøglen $x.\text{KEY}$ gemt i $A[i]$. Vi er ofte interesseret at gemme noget ekstra information (de såkaldte 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 af satellitdata i $O(1)$ tid kun givet indeks i , mens man stadig bibeholder køretiden for de normale hoboperationer.

5 Hobegenskaber

Lad T være et perfekt binært træ af højde h . Løs følgende opgaver:

- 5.1 Vis at antallet af knuder i T er $n = 2^{h+1} - 1$. *Hint*: vis, at antallet af knuder i T er $n = 1 + 2 + 4 + \dots + 2^h$, og skriv tallet som binært tal.
- 5.2 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 arrayrepræsentationen. Løs følgende opgaver:

- † Implementer operationerne INSERT og EXTRACTMAX .

7 Delsummer

Lad $A[0..n-1]$ være et array af heltal. Vi er interesseret i følgende operationer på A :

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

Løs følgende opgaver:

- 7.1 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.