

## Ugeseddel 7

### Litteratur:

- CLRS intro part II
- CLRS kapitel 6 (ignorer tekst om master theorem)
- CLRS B.5, kun i den udstrækning som det er nødvendigt for at forstå kapitel 6 og løse opgaver.
- CLRS formel A.8 bruges til Build\_max\_heap analyse
- CLRS C.1 om permutationer. Kun som støtte til afsnit 8.1
- CLRS kapitel 8 (dog ikke analysen af bucket sort startende på side 202. Dertil er side 199 kursorisk.)

### Noter:

- 7.1 prioritetskøer noter CLRS 6 og B5
- 7.2-3 sortering i linear tid
- 7.3 Nedre grænse for sammenligningsbaseret sortering

### Mål for ugen:

- Prioritetskøer og hobe.
- Sorteringsmetoder.
- Sortering i visse tilfælde kræver mere end lineær tid, og i andre tilfælde kan løses i lineær tid.

### Mål for ugen:

- Tirsdag: Prioritetskøer og hobe
- Spørgetimen tirsdag: Mere om matematikken bag hobe
- Torsdag: Sortering i lineær tid og nedre grænse for sammenligningsbaseret sortering
- Spørgetimen torsdag: Mest om CLRS kapitel 8, Stephen.

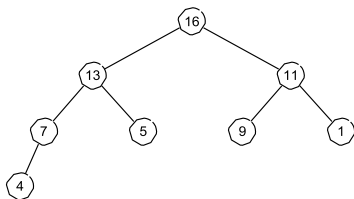
### Opgaver tirsdag:

Opgaver markeret med (ekstra) kan gemmes til resten af opgaverne er regnet. I bunden af ugesedlen finder du nogle svære opgaver markeret med stjerne hvis du mangler udfordring eller er færdig med dagens opgaver.

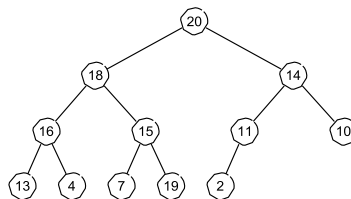
Pas på: Opgaver markeret med "\*" er svære, "\*\*" er meget svære, og "\*\*\*" har du ikke en chance for at løse.

### 1. Hobegenskaber og håndkøring Løs følgende opgaver.

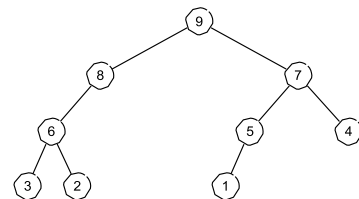
#### 1.1. Hvilke af følgende træer er en hob?



(a)



(b)



(c)

1.2 Hvilke af følgende tabeller er en hob? Index 0 som ikke bruges er markeret med  $-$ .

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

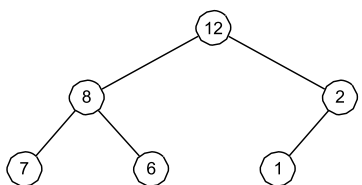
1.3 Lad  $S = (4, 8, 11, 5, 21, *, 2, *)$  være en sekvens af operationer hvor alle tal svarer til en indsættelse af tallet og  $*$  svarer til en EXTRACTMAX operation. Startende med en tom hob  $H$ , vis hvordan  $H$  ser ud efter hver operation i  $S$ .

1.4 [\*] Vis at INSERT, EXTRACTMAX og INCREASEKEY bibeholder hoborden.

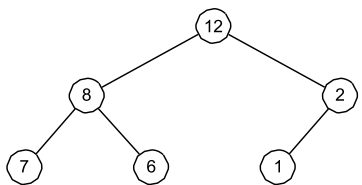
1.5 [\*] CLRS 6.5-9.

## 2. Hobetegninger

2.1 Tegn hvordan nedenstående max-hob ser ud efter indsættelse af et element med nøgle 4



2.2 Tegn hvordan nedenstående max-hob ser ud efter en EXTRACT-MAX operation



## 3. Hobe

3.1. CLRS 6.1-4

3.2. CLRS 6.1-5

3.3. CLRS 6.1-6 (ekstra)

3.4. CLRS 6.1-7 (ekstra) *Hint*: Tænk over hvor i tabellen den sidste knude som har børn ligger.

## 4. Prioritetskøer (ekstra)

4.1. CLRS 6.5-1

4.2. CLRS 6.5-2

4.3. CLRS 6.5-3

## 5. Max Heapify

5.1. CLRS 6.2-1

5.2. CLRS 6.2-3

5.3. CLRS 6.2-4

## 6. Build max-heap

6.1. CLRS 6.3-1

6.2. CLRS 6.3-2

6.3. CLRS 6-1

7. **Prioritetspolitik (ekstra)** 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 med lavest indkomst og deportere dem. Specifik skal systemet understøtte følgende operationer.

- $\text{INDSÆT}(c, i)$ : indsæt person med cpr. nr.  $c$  og årlig indkomst  $i$  i systemet.
- $\text{SLET LAVESTE INDKOMST}()$ : fjern og returner (deporter) person med laveste indkomst.

7.1 Foreslå en effektiv datastruktur  $M$  til systemet.

7.2 [\*] Antiteknokraterne, et andet ekstremt, får senere magten og indføre et system hvor dem med de højeste indkomster deporteres. Landets statslige IT leverandør laver derfor en ny datastruktur  $H$  der understøtter dette system. Senere overtager Fjolleristerne magten. De vil have et system  $M$  der kan deportere den person som har den midterste (media) indkomst. Vis hvorledes  $M$  kan implementeres vha.  $H$  og  $M$ .

## 8. Hobegenskaber [\*] (Matematikken til denne opgave vil blive gennemgået til forelæsning eller spørgetime, gem evt. opgaven til da)

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

- 8.1. Vis at antallet af knuder i  $T$  er  $n = 2^{h+1} - 1$ . *Hint*: vi har at  $n = 1 + 2 + 4 + \dots + 2^h$ . Multiplicer summen med 2 og træk summen fra. Eller tænk på summen i binære repræsentation.
- 8.2. 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  $S - S/2$ .
- 8.3. CLRS 6.1-1
- 8.4. CLRS 6.1-2
- 8.5. [\*] CLRS 6.3-3

## 9. Heapsort

- 9.1. CLRS 6.4-1
- 9.2. CLRS 6.4-3

## Opgaver torsdag:

### 1. Counting sort

- 1.1. CLRS 8.2-1
- 1.2. CLRS 8.2-2
- 1.3. CLRS 8.2-3
- 1.4. CLRS 8.2-4 *Hint*: Brug tabellen  $C$  fra counting sort.

### 2. Radix sort

- 2.1. CLRS 8.3-1
- 2.2. CLRS 8.3-4

### 3. Bucket sort

- 3.1. CLRS 8.4-1
- 3.2. CLRS 8.4-2

### 4. Sortering i forskellige situationer

- 4.1. Antag at  $n$  personer skal sorteres efter deres højde hvor højden er angivet i antal hele cm. Hvilken algoritme vil du bruge og hvad er køretiden?
- 4.2. Antag at  $n$  heltal fra  $1 \dots n^{10}$  skal sorteres. Hvilken algoritme vil du bruge og hvad er køretiden?
- 4.3. Antag du skal sortere  $n$  objekter. Du kan ikke sammenligne objekterne direkte. Men du kan spørge et orakel  $A$  hvilket af to objekter er mindst. For hvert spørgsmål bliver du opkrævet  $O(\log \log n)$  guldmønter. Hvilken algoritme vil du bruge og hvor mange guldmønter skal du aflevere til oraklet?

### 5. Sorting in place in linear time

- 5.1. CLRS 8-2 a
- 5.2. CLRS 8-2 b
- 5.3. CLRS 8-2 c

## 6. Comparison sortering

### 6.1. CLRS 8.1-1

7. Husk at se på opgave 8 fra tirsdag hvis du ikke har løst den endnu.

### Stjerneopgaver (svære ekstraopgaver):

**1 Prioritetskøoperationer** Vi vil gerne tilføje nogle operationer til vores (maks) prioritetskø. Vi er interesseret i at tilføje følgende operationer. I nedestående er  $x$  og  $y$  objekter.  $x.key$  og  $y.key$  er deres nøgler i prioritetskøen (objekterne kan have andre satellitdata). Hvis en implementation af prioritetskøen sker som i CLRS kapitel 6 kan tabellen være en tabel af objekter – og ikke tal. Her kan  $x$  og  $y$  så tænkes at have en attribut  $i$ , således at  $x.i$  og  $y.i$  angiver deres plads (index) i tabellen.

- **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( $k$ )**: returner de elementer i hoben med nøgle  $\geq k$ .
- **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.

**1.1** Udvid prioritetskøen til at understøtte **REMOVELARGEST( $m$ )** i  $O(m \log n)$  tid.

**1.2** Udvid prioritetskøen til at understøtte **DELETE** og **FUSION** i  $O(\log n)$  tid.

**1.3** [\*] Udvid prioritetskøen til at understøtte **FINDLARGEST( $k$ )** i  $O(m)$  tid, hvor  $m$  er antallet af elementer med nøgle  $\geq k$ .

**1.4** [\*\*] Udvid prioritetskøen til også at understøtte **EXTRACTMIN** i  $O(\log n)$  tid.

**2. Delsummer [S]** 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 (Det er kun i opgave 11.3 man muligvis har brug for en hoblignende struktur).

**2.1** Giv en datastruktur, der understøtter **SUM** i  $O(1)$  tid og bruger  $O(n^2)$  plads.

**2.2** [\*] Giv en datastruktur, der understøtter **SUM** i  $O(1)$  tid og bruger  $O(n)$  plads.

**2.3** [\*\*\*] Giv en datastruktur, der understøtter både **SUM** og **CHANGE** i  $O(\log n)$  tid og bruger  $O(n)$  plads.

## 3. $d$ -nære hobe

### 3.1 [\*] CLRS 6-2

### Bemærkninger:

Nogle opgaver er stærkt inspireret af opgaver stillet af Philip Bille og Inge Li Gørtz i kurset Algoritmer og Datastrukturer, på DTU, <http://www2.compute.dtu.dk/courses/02105+02326/2015/#generelinfo>