

# Introduktion til datastrukturer

**Materialer** Introduction to Algorithms (CLRS): introduktion til Del III + kap. 10.

## Opgaver

### 1 Stakke og køer

- 1.1 [w] Vis udførelsen af sekvensen PUSH(4), PUSH(1), PUSH(3), POP, PUSH(8), POP på en stak, som starter med at være tom, gemt i et array med kapacitet 6.
- 1.2 Vis hvordan man kan implementere to stakke i et enkelt array  $A[0 \dots N-1]$  således at ingen af stakkene flyder over så længe det samlede antal elementer er under  $N$ . PUSH og POP skal tage  $O(1)$  tid.
- 1.3 [w] Vis udførelsen af sekvensen ENQUEUE(4), ENQUEUE(1), ENQUEUE(3), DEQUEUE, ENQUEUE(8), DEQUEUE i en kø, som starter med at være tom, gemt i et array med kapacitet 6.
- 1.4 [\*] Vis hvordan man kan implementere en kø ved hjælp af to stakke (og ingen andre datastrukturer). Analyser køretiden af køoperationerne.

### 2 Opgave 5.1 fra eksamen 2011

Lad  $S$  være en stak. Udfør følgende operationer fra venstre mod højre. Bogstavet  $i$  betyder PUSH( $S, i$ ) og  $*$  betyder POP

D \* T U \* \* I N \* F O R \* M \* A T I K

Hvilke af følgende sekvenser af bogstaver svarer til de bogstaver, der bliver returneret af POP( $S$ )?

- D U T I R M
- D U T N R M K I T A O F
- D U T N R M
- D T U I N F
- D U T N R M I T A O F
- D U T N O M

### 3 Algoritmer på hægtede lister

Kig på algoritmerne `FOO` og `BAR` og den hægtede liste nedenfor. Løs følgende opgaver:

- 3.1 Håndkør `FOO(head)`.
- 3.2 Forklar hvad `FOO` gør.
- 3.3 Håndkør `BAR(head, 0)`.
- 3.4 Forklar hvad `BAR` gør.

```
FOO(head)
x = head
c = 0
while x ≠ null do
  x = x.next
  c = c + 1
end while
return c
```

```
BAR(x, s)
if x == null then
  return s
else
  return BAR(x.next, s + x.key)
end if
```



#### 4 Implementation af hægtede lister

Antag  $x$  er element i en enkelt-hægtet liste som beskrevet til forelæsningen. Løs følgende opgaver:

- 4.1 Antag  $x$  ikke er det sidste element i listen. Hvad effekten af den følgende kodestump?

```
x.next = x.next.next;
```

- 4.2 Lad  $t$  være et nyt element der ikke er i listen i forvejen. Hvad er effekten af følgende kodestump?

```
t.next = x.next;  
x.next = t;
```

- 4.3 Hvorfor gør følgende kodestump ikke det samme som forrige delopgave?

```
x.next = t;  
t.next = x.next;
```

#### 5 Implementation af stakke og køer

Løs følgende opgaver:

- 5.1 † Implementér en stak der kan indeholde heltal ved hjælp af en enkelt-hægtet liste.  
5.2 † Implementér en kø der kan indeholde heltal ved hjælp af en enkelt-hægtet liste.

#### 6 Sorterede hægtede lister

Lad  $L$  være en enkelt-hægtet liste bestående af  $n$  heltal i sorteret rækkefølge. Løs følgende opgaver:

- 6.1 Giv en algoritme til at indsætte et nyt tal i  $L$  således at listen bliver ved med at være sorteret. Analyser din algoritme ved brug af  $\Theta$ -notation.  
6.2 En ven foreslår at man kan benytte binær søgning til hurtigere at søge i en sorteret hægtet liste. Har vennen ret?

#### 7 Listevending

Giv en algoritme til at "vende" en enkelt-hægtet liste om, dvs. producere en enkelt-hægtet liste bestående af de samme elementer i omvendt rækkefølge. Din algoritme skal køre i  $\Theta(n)$  tid og ikke bruge mere end konstant plads udover pladsen brugt af selve listen.

#### 8 Død ved lampe

32 fanger er fængslet på livstid i isolation. Fængslset indeholder 32 celler og et forhørskammer, forhørskammeret indeholder  $k$  lamper med lyskontakter, og fangerne får ikke mulighed for at se eller høre hinanden efter fængsling. Den onde fængselsvagt foreslår følgende spøg: Hver dag vælger fængselsvagten en tilfældig fange (ved at vælge et tilfældigt tal mellem 1 og 32), som kommer i forhørskammeret, hvor de har mulighed for både at observere og at tænde eller slukke for hver af de  $k$  lamper. Fængselsvagten rører ikke ved lamperne, og til start er de alle slukkede.

Til enhver tid må enhver fange sige: "Alle 32 fanger har været i forhørskammeret." Hvis udsagnet er korrekt, bliver alle sat fri, hvis udsagnet er forkert, bliver alle henrettet. Inden de blev fængslet havde fangerne heldigvis rådført sig med hinanden og lagt en god strategi for, hvordan de skulle bære sig ad med at vinde over vagten.

8.1 Findes der en strategi så fangerne kan vinde over fængselsvagten hvis der er  $k = 32$  lyskontakter?

8.2 Hvis  $k = 5$ ?

8.3 **[\*\*]** Hvis  $k = 1$ ?

## 9 Dynamiske arrays og stakke

Vi er interesseret i at implementere en stak ved hjælp af dynamiske arrays uden at fastsætte en maksimal størrelse på arrayet til at starte med. Løs følgende opgaver:

9.1 **[\*]** Generalisér dynamiske arrays til også at kunne håndtere stakke der ”skrumper” undervejs (dvs. understøtter både PUSH og POP operationer). Køretiden af enhver sekvens af  $n$  operationer skal være  $\Theta(n)$  og til ethvert tidspunkt skal din løsning bruge lineær plads i antallet af elementer i stakken.

9.2 **[\*\*]** Vis hvordan man med dynamiske arrays kan opnå

- $O(1)$  tid per stakoperation og
- pladsforbrug der er lineært i det nuværende antal af elementer i stakken

Kig kun på stakke der vokser og ignorer ”skrump”.

Hint: Tænk på hvordan man kan fordele arbejdet blandt operationerne.