Institut for Matematik og Datalogi
Syddansk Universitet, Odense

27. september 2018
LMF/RF

Eksaminatorier DM534

Studiegrupperne handler om at bruge hinanden til at arbejde med stoffet og sammen forsøge at løse ugens opgaver inden eksaminatorietimerne. I denne uge er mange af opgaverne formuleret som spørgsmål, hvilket naturligt kan danne baggrund for diskussion i en studiegruppe.

Husk at læse de relevante slides før du/I forsøger at løse en opgave.

Uge 40

1. Er nedenstående en algoritme?

   $i = 0$
   While $i \neq 5$
   $\quad i = i + 2$

2. Betragt listen $L = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]$. I nedenstående spørgsmål tæller vi sammenligninger, som involverer elementer i listen.

   (a) Hvor mange sammenligninger foretages der med SequentialSearch$(L, 7)$?
   (b) Hvor mange sammenligninger foretages der med BinarySearch$(L, 7)$?

   Antag nu, at L indeholder 10.000 elementer.

   (c) Hvor mange sammenligninger foretager man i værste tilfælde med en sekventiel søgning i L?
   (d) Hvor mange sammenligninger foretager man i værste tilfælde med en binær søgning i L?

3. Udfyld de manglende felter (undtagen øverste række) i tabellen på side 10 i forelæsningsnoterne om algoritmer.

4. Husk på algoritmerne til, ciffer for ciffer, at addere eller gange to tal i hånden. Hvis du ikke helt kan huske dem, er her et eksempel:

$$\begin{array}{rr} & 321 \\ + & 281 \\ \hline & 602 \end{array} \qquad \begin{array}{r} 321 \times 281 \\ \hline 281 \\ 562 \\ 843 \\ \hline 90201 \end{array}$$

   (a) Hvad er køretiden for at addere to tal med n cifre hver? Hvad er den karakteristiske operation?

   (b) Hvad er køretiden for at gange to tal med n cifre hver? Hvad er den karakteristiske operation?

5. Hvilke af følgende udsagn er sande?

   (a) $n \in O(n)$

   (b) $2n + 5 \in O(n)$

   (c) $\sqrt{n} - \log(n) \in O(n)$

   (d) $(\log(n))^2 \in O(n \log n)$

   (e) $n^2 \in O(n)$

   (f) $n \in O(n^2)$

   (g) $n \log(n) \in O(n^2)$

   (h) $n \log(n) \in O(n)$

   (i) $3n^2 + 2n + 1 \in O(n^2)$

   (j) $3n^2 + 2n + 1 \in O(n)$

6. Betragt følgende algoritme til at finde det mindste tal i listen L.

```
Min(L)
n = L.length
min = L[1]
For i = 2 to n
    If L[i] < min
        min = L[i]
Return min
```

(a) Hvad er algoritmens køretid?

(b) Opskriv en løkke-invariant for algoritmen, og bevis, at den altid finder det mindste element i L.

(c) Omskriv algoritmen, så den bruger en while-løkke i stedet for en for-løkke.

(d) Bemærk, at algoritmen er iterativ. Skriv en rekursiv version af algoritmen.

7. Angiv køretiden for hver af nedenstående algoritmer.

(a)
$i = 1$
While $i \leq n$
    $i = i + 1$

(b)
$i = 1$
While $i \leq n$
    $i = i * 3$

(c)
$i = 1$
For $k = 1$ **to** n
    For $l = 1$ **to** n
        $i = i + k + l$

8. Betragt følgende algoritme.

NUMBERS(n)
print n
If $n < 3$
    NUMBERS($n + 1$)
print n

Hvilken talfølge skriver NUMBERS(1)?

9. Fibonacci-tallene er defineret således:

$$
\begin{aligned}
f_0 &= 0 \\
f_1 &= 1 \\
f_i &= f_{i-1} + f_{i-2}, \text{ for } i \geq 2
\end{aligned}
$$

Skriv en iterativ og en rekursiv algoritme, som beregner det i'te fibonaccital. Implementer begge algoritmer. Hvilken version er hurtigst? Hvad kan en evt. forskel i køretid skyldes?