Institut for Matematik og Datalogi
Syddansk Universitet, Odense

30. september 2003
finie02@imada.sdu.dk

DM02 – Ugeseddel 4

Øvelsesopgaver 23/9, 24/9 og 25/9

1. Cormen et al. 2.3-3
   Brug matematisk induktion til at vise at når n er en potens af 2, dvs. $n = 2^k$, så har følgende rekursionsligning

$$T(n) = \begin{cases} 2 & , n = 2 \\ 2T(\frac{n}{2}) + n & , n = 2^k, k > 1 \end{cases}$$

   løsningen $T(n) = nlg(n)$

   Basis $n = 2 = 2^1$, dvs. $k = 1$

$$T(2) = 2 = 2 * 1 = 2lg(2)$$

   Vi ser at basistilfældet holder.

   Induktion $n = 2^k$, $k \geq 1$
   Antager $T(n) = nlgn$ for $k \geq 1$.
   Viser for $k + 1$:

$$\begin{aligned} T(n) &= 2T(\frac{n}{2}) + n \\ (n = 2^{k+1}) &= 2T\left(\frac{2^{k+1}}{2}\right) + 2^{k+1} \\ &= 2T(2^k) + 2^{k+1} \\ \text{(ved ind. ant.)} &= 2(2^k lg(2^k)) + 2^{k+1} \\ &= 2^{k+1} lg(2^k) + 2^{k+1} \\ &= 2^{k+1} klg(2) + 2^{k+1} \\ &= 2^{k+1} klg(2) + 2^{k+1} \\ &= 2^{k+1} k + 2^{k+1} \\ &= 2^{k+1}(k + 1) \\ &= 2^{k+1} lg(2^{k+1}) \end{aligned}$$

   Altså har vi vist at $T(n) = nlg(n)$ for alle $n = 2^k$ hvor $k > 0$.

   Det eneste specielle ved dette induktionsbevis sammenlignet med tidligere opgaver, er at vi her laver induktionen over k og ikke over n som vi hidtil har været vant til.

2. 2.3-5
   Skriv en iterativ eller rekursiv pseudokode for binær søgning.
   Argumentér for at worstcase køretiden er $\Theta(lg(n))$.

   Først et eksempel på en rekursiv implementering:

```
BinarySearch(A, 0, Length[A], key) \\første kald til funktionen

BinarySearch(A, start, end, key)
  if start == end
    if A[start] == key
      return start
    else
      return -1

  m := floor[(end + start)/2]

  if key > A[m]
    return BinarySearch(A, m+1, end, key)
  else
    return BinarySearch(A, start, m, key)
```

og så et eksempel på en iterativ implementering:

```
BinarySearch(A, key)
  start := 0
  end := Length[A]

  while (end-start > 0)
    m := Floor[(end+start)/2]
    if (key > A[m])
      start := m+1
    else
      end := m

  if ( A[start] == key )
    return start
  else
    return -1
```

Bemærk at vi her bruger værdien -1 som en returværdi der angiver at nøglen 'key' ikke findes i arrayet.

Køretid
Vi opskriver rekursionsligningen for worst case:

$$T(n) = \begin{cases} 1 & ,n = 1 \\ T(\lceil \frac{n}{2} \rceil) + \Theta(1) & ,ellers \end{cases}$$

Vi kan se at rekursionen ender i basistilfældet når vi har delt n i to så mange gange at n bliver 1. Spørgsmålet er nu, hvormange gange kan vi højst dele et array i to?
Svaret er $log_2(n)$

Formelt kan vi beregne køretiden for rekursionsligningen ved at bruge Master Theorem (Cormen s. 62, 73), da vi kan genkende $T(n)$ som en ligning på formen

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

Hvor $a = 1$, $b = 2$, $f(n) = \Theta(1)$.
Vi udregner da $n^{\log b(a)}$:

$$n^{\log_b(a)} = n^{\log_2(1)} = n^0 = 1$$

og herfra giver Master Theorem resultatet:

$$\Rightarrow f(n) \in \Theta(n^{\log_b(a)}) = \Theta(1)$$

$$\Rightarrow T(n) \in \Theta(n^{\log_b(a)} lg(n)) = \Theta(lg(n))$$

3. 2.3-6.
(udeladt med vilje)

4. Bevis, at $n! \in \omega(2^n)$ og $n! \in o(n^n)$.
Det er lettest at udregne grænseværdierne for $\frac{f(n)}{g(n)}$ som og tjekke resultatet med definitionerne for asymptotisk notation beskrevet på ugeseddel 2.
Først viser vi at $n! \in \omega(2^n)$:

$$\lim_{n\to\infty} \frac{n!}{2^n} = \lim_{n\to\infty} \frac{n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1}{2 \cdot 2 \cdot \ldots \cdot 2 \cdot 2} = \infty$$

Dernæst $n! \in o(n^n)$:

$$\lim_{n\to\infty} \frac{n!}{2^n} = \lim_{n\to\infty} \frac{n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1}{n \cdot n \cdot \ldots \cdot n \cdot n} = 0$$

5. Cormen et al. 3.1-7.
Bevis at $o(g(n)) \cap \omega(g(n)) = \emptyset$.
Bruger definitioner på o og ω fra Cormen s. 48:

$$o(g(n)) = \left\{ f(n) : \forall c > 0 : \exists n_0 > 0 : 0 \leq f(n) < cg(n), \forall n \geq n_0 \right\}$$

$$\omega(g(n)) = \left\{ f(n) : \forall c > 0 : \exists n_0 > 0 : 0 \leq cg(n) < f(n), \forall n \geq n_0 \right\}$$

Vi antager til modstrid at vi har en funktion $f(n)$ som tilhører både $o(g(n))$ og $\omega(g(n))$

$$f(n) \in o(g(n)) \Leftrightarrow 0 \leq f(n) < cg(n), \forall c_1, n \geq n_1$$

$$f(n) \in \omega(g(n)) \Leftrightarrow 0 \leq cg(n) < f(n), \forall c_2, n \geq n_2$$

Altså skal der gælde, for $n \geq max(n_1, n_2)$ i tilfældet for $c_1 = c_2$:

$$c_1 g(n) < f(n) < c_1 g(n)$$

Altså har vi en **Modstrid**, da f(n) ikke både kan være skarpt større og skarpt mindre end c g(n).
Altså har vi vist at mængden er tom.

Institut for Matematik og Datalogi
Syddansk Universitet, Odense 30. september 2003
finie02@imada.sdu.dk

6. Cormen et al. 3-4 a, d, f, g, h.
   Bevis eller modbevis hvert af følgende udsagn:
   a)$f(n) = O(g(n)) \Rightarrow g(n) = O(f(n))$
   d)$f(n) = O(g(n)) \Rightarrow 2^{f(n)} = O(2^{g(n)})$
   f)$f(n) = O(g(n)) \Rightarrow g(n) = \Omega(f(n))$
   g)$f(n) = \Theta(f(\frac{n}{2}))$
   h)$f(n) + o(f(n)) \in \Theta(f(n))$

   Enten skal bevises vha. definitionerne i Cormen (eller fra ugeseddel 2) at udsagnene er eviggyldige, eller også skal der findes et modeksempel.

   a) Modstridseksempel: $f(n) = n$ og $g(n) = n^2$
   Udregner grænseværdierne

$$\lim_{n\to\infty} \frac{n}{n^2} = 0 < \infty \Rightarrow n \in O(n^2)$$

$$\lim_{n\to\infty} \frac{n^2}{n} = \infty \Rightarrow n^2 \notin O(n)$$

   Altså er udsagnet FALSK.

   d) FALSK

   f) SANDT

   g) FALSK

   h) SANDT