Diskret Matematik og Algoritmer
Ugeopgave 3

Mads Pontoppidan Haderup (xjr983), Ken Kjøller Andersen (zcj256)
og Christian Handest (lzp959)

30. september 2018

1 DEL 1

Afgør for hver af følgende 4 underopgaver, om den ene funktion er af lavere størrelsesorden end den anden, eller om de to funktioner er af samme størrelsesorden. Argumenter for konklusionerne ud fra notesættets regler O1–O9 eller S1–S9.

1.1 DEL 1A

$$f(n) = \log_2(n) \ og \ g(n) = n^2 + 2n$$

Det undersøges, om S3 kan benyttes for at finde størrelsesordenen for $f(n)$ og $g(n)$.

> S3: "$log_a(x)$ er af lavere størrelsesorden end x^b (a $>$ 1, b $>$ 0)"

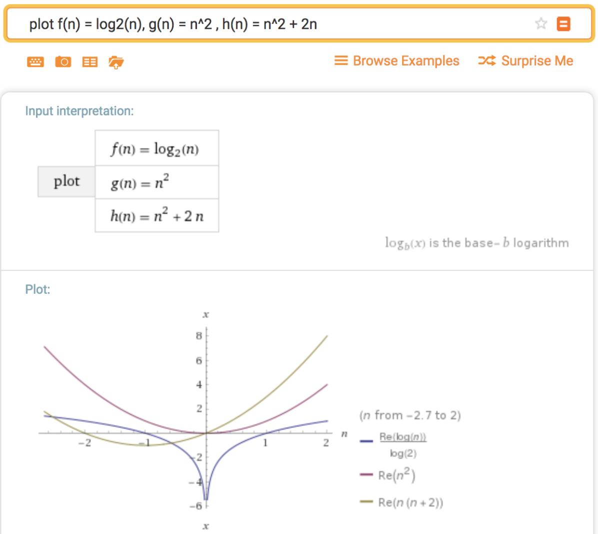
For at benytte S3 skal $f(n)$ og $g(n)$ opfylde betingelserne for at $a > 1$ og $b > 0$.
Disse betingelser er opfyldte, da $a = 2$ og $b = 2$.

Derudover skal det undersøges, om $g(n) = n^2 + 2n$ kan skrives på formen $g(n) = n^2$.
Da det allerede antages, at $log_2(n) < n^2$, vil $n^2 + 2n$ kun forøge afstanden mellem $g(n)$ og $f(n)$. Netop derfor kan vi benytte S1 som siger:

> S1: "Hvis f(x) er af lavere størrelsesorden end g(x), og g(x) er af lavere størrelsesorden end h(x), så er f(x) af lavere størrelsesorden end h(x)."

I vores tilfælde har vi så at $f(n) = log_2(n)$, $g(n) = n^2$ og $h(n) = n^2 + 2n$.

Følgende graf illustrerer hvordan $f(n) < g(n)$, men også hvordan $g(n)$'s oprindelige $+2n$ er overflødig i sammenligningen mellem $f(n)$ og $g(n)$, for så vidt angår afgørelse af størrelsesordnen mellem de to funktioner.



Til sidst kan S3 benyttes til afgøre størrelsesordenen mellem $f(n) = log_2(n)$ og $g(n) = n^2$.

Størrelsesordenen mellem $g(n)$ og $f(n)$ er:

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

1.2 DEL 1B

$$f(n) = n\sqrt{n} \ og \ g(n) = nlog_2(n) + 100n$$

f(n) kan skrives som

$$f(n) = n^{\frac{3}{2}}$$

g(n) skal forkortes, så vi ser på størrelsesordenen mellem

$$nlog_2(n) \ og \ 100n$$

Vi ved efter S2, at en konstant 100 er af lavere størrelsesorden end $log_2(n)$ for alle a $>$ 0. Og fra S8 ved vi, at $n100$ er af lavere end $nlog_2(n)$. Derfor kan vi opskrive g(n) på følgende måde: $g(n) = log_2(n)$. Efter S3 kan vi til sidst finde, at g(n) er af lavere størrelsesorden end f(n).

Størrelsesordenen mellem $f(n)$ og $g(n)$ er

$$f(n) > g(n)$$

1.3 DEL 1C

$$f(n) = 2^{\log_3(n)} \ og \ g(n) = \frac{2^n}{n^{100}}$$

Vi ved fra S6, at

"x^a er af lavere størrelsesorden end b^x (a $>$ 1, b $>$ 1)"

Da 2^n er af lavere størrelsesorden end n^{100} kan vi vide, at g(n) går mod 0. Vi kan også se, at f(n) går mod uendelig. Vi kan skrive følgende:

$$\lim_{n\to\infty} 2^{\log_3(n)} = \infty$$

$$\lim_{n\to\infty} \frac{2^n}{n^{100}} = 0$$

Størrelsesordenen mellem $f(n)$ og $g(n)$ er

1.4 DEL 1D

$$f(n) = (n + log_2(n))^2 \ og \ g(n) = n^2$$

Vi kan dividere med kvadratroden i begge funktioner, da det ikke vil ændre størrelsesforholdet mellem de to funktioner.

$$f(n) = \sqrt{(n + \log_2(n))^2} \ og \ g(n) = \sqrt{n^2}$$

Vi har så, at

$$f(n) = n + \log_2(n) \ og \ g(n) = n^2$$

Herefter skal $f(n)$ forkortes, sådan at vi finder ud af hvilken størrelsesorden n har i forhold til $\log_2(n)$. Efter S3 er $\log_2(n)$ af lavere størrelsesorden end n^1, hvorfor vi kan skrive $f(n)$ og $g(n)$ således:

$$f(n) = n \ og \ g(n) = n^2$$

Efter S5 er n^1 af lavere størrelsesorden end n^2, hvorfor vi har, at:

Størrelsesordenen mellem $f(n)$ og $g(n)$ er

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

2 DEL 2

Vi betragter fire følger givet ved henholdsvis
$a_1 = 2$, $a_2 = 200$, $a_n = a_{n-2}$ for n ≥ 3
$b_1 = 2$, $b_n = b_{n-1} + 2$ for n ≥ 2
$c_1 = 1$, $c_n = nc_{n-1}$ for n ≥ 2
$d_n = log_2(c_n)$ hvor c_n er defineret som oven over og n ≥ 1.

2.1 DEL 2A

For hver af de 3 rekursive følger a_n, b_n og c_n, find et eksplicit udtryk for det n't led. Det vil sige, find funktioner $f_a(n)$, $f_b(n)$, $f_c(n)$ således $a_n = f_a(n)$, $b_n = f_b(n)$, $c_n = f_c(n)$.

$a_n = a_{n-2}$ for n ≥ 3
$a_1 = 2$
$a_2 = 200$
$a_3 = a_{3-2} = a_1 = 2$
$a_4 = a_{4-2} = a_2 = 200$
$a_5 = a_{5-2} = a_2 = 2$
$a_6 = a_{6-2} = a_4 = 200$

Af ovenstående ses et mønster, hvor der i tilfælde af det n'te led i følgen er et ulige tal, er det korresponderende indekstal 2. Omvendt, hvis det n'te led i følgen er et lige tal er indekstallet 200.

Ud fra dette mønster kan vi danne følgende funktion:
$$f_a(n) = \begin{cases} 2 \text{ hvis } n - 2\lfloor \frac{n}{2} \rfloor = 1 \\ 200 \text{ hvis } n - 2\lfloor \frac{n}{2} \rfloor = 0 \end{cases}$$

$b_n = b_{n-1} + 2$ for n ≥ 2
$b_1 = 2$
$b_2 = b_{2-1} + 2 = b_1 + 2 = 2 + 2 = 4$
$b_3 = b_{3-1} + 2 = b_2 + 2 = 4 + 2 = 6$
$b_4 = b_{4-1} + 2 = b_3 + 2 = 6 + 2 = 8$

Ud fra dette mønster kan vi danne følgende funktion: $f_b(n) = 2n$

$c_n = nc_{n-1}$ for n ≥ 2
$c_1 = 1$
$c_2 = 2c_{2-1} = 2c_1 = 2 * 1 = 2$
$c_3 = 3c_{3-1} = 3c_2 = 3 * 2 = 6$
$c_4 = 4c_{4-1} = 4c_3 = 4 * 6 = 24$
$c_5 = 5c_{5-1} = 5c_4 = 5 * 24 = 120$

Ovenstående følge er også kendt som fakultetsfølgen. Derfor bliver $f_c(n) = n!$

2.2 DEL 2B

Husk på, at vi siger (a_n) er $\Theta(b_n)$, hvis følgen (a_n) er af samme størrelsesorden som følgen (b_n). For hver af de 3 følger (a_n), (b_n) og (d_n), find en tilsvarende størrelsesorden fra nedensåtende liste der korrekt beskriver følgens størrelsesorden:
$\Theta(1), \Theta(n), \Theta(nlog_2(n)), \Theta(n^2), \Theta(2^n)$
Argumentér for konklusionerne ud fra notesættets sætninger og regler (O1-O9, S1-S9).

$$f_a(n) = \begin{cases} 2 \text{ hvis } n - 2\lfloor \frac{n}{2} \rfloor = 1 \\ 200 \text{ hvis } n - 2\lfloor \frac{n}{2} \rfloor = 0 \end{cases}$$

Ovenstående funktion er en relativt simpel funktion, hvor outputtet kun kan være 2 eller 200. Dermed giver et input, hvor n $\geq$ 1 et øjeblikkeligt output uden yderligere bearbejdning, hvorfor der kun er 1 trin for denne funktion. Af den årsag er $\Theta(1)$ af samme størrelsesorden som $f_a(n)$.

Sætning S7 foreskriver, at cf(x) er af samme størrelsesorden som f(x) når c $\neq$ 0. Funktionen $f_b(n) = 2n$, hvor c = 2 og derfor ikke 0. Dermed er $\Theta(n)$ af samme størrelsesorden som $f_b(n)$

$d_n = log_2(c_n) = log_2(n!)$ hvor c_n er defineret som oven over og n $\geq$ 1.
Sætning 17 foreskriver, at $log_2 n!$ er $\Theta(nlog_2(n))$. Dermed er $\Theta(nlog_2(n))$ af samme størelsesorden som $log_2(c_n)$.

3 DEL 3

Find udtryk for køretid (antal af udførte instruktioner) af COMPUTE(n) ved input n $\geq$ 1 som en rekursiv følge (t_n). Du kan antage at hver linje af programmet svarer til en instruktion når du tæller instruktioner. Hvilken af Θ-notationerne fra Eq. (1) beskriver køretid af COMPUTE(n)?

	Times
function Compute(n)	n
if n = 1 then	n
return n	1
else	n-1
return n*Compute(n-1)	n-1

T(n) = n + n + 1 + (n - 1) + (n - 1) = 2(n) + 2(n-1) + 1 = O(n)