

Løsningsforslag til kapittel 3 i
Introduction to Algorithms (Cormen et al.)

Til bruk i emnet TDT4120 Algoritmer og Datastrukturer

Åsmund Eldhuset

Inneholder løsninger av følgende oppgaver:

3.1: 1 til 7

3.2: 1

3: 2, 4

Kapittel 3

Seksjon 3.1 - Treningsoppgaver

3.1-1

La oss anta at $f(n) \geq g(n)$ for en eller annen n . Da gjelder

$$\begin{aligned}f(n) &= \max(f(n), g(n)) \\g(n) &\leq \max(f(n), g(n))\end{aligned}$$

og dermed

$$\begin{aligned}f(n) + g(n) &\leq \max(f(n), g(n)) + \max(f(n), g(n)) \leq 2 \max(f(n), g(n)) \\ \frac{1}{2}(f(n) + g(n)) &\leq \max(f(n), g(n)).\end{aligned}$$

Siden begge funksjonene er ikke-negative, kan vi være sikre på at $0 \leq f(n) \leq f(n) + g(n)$.

$$\max(f(n), g(n)) = f(n) \leq f(n) + g(n)$$

og dermed

$$0 \leq \frac{1}{2}(f(n) + g(n)) \leq \max(f(n), g(n)) \leq f(n) + g(n)$$

Vi kan gjøre tilsvarende (og ende opp med samme resultat) dersom $f(n) < g(n)$ for en eller annen n . Dermed gjelder uttrykket over uansett verdien av n . Betingelsen på side 42 er da oppfylt (siden dette gjelder for alle verdier av n , kan n_0 velges lik hva som helst), og vi kan konkludere med at $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

3.1-2

I følge definisjonen av Θ må vi finne tre konstanter c_1 , c_2 og n_0 som er slik at $0 \leq c_1 n^b \leq (n+a)^b \leq c_2 n^b$ for alle $n \geq n_0$. La oss dele opp i tre tilfeller, avhengig av verdien av a :

$a > 0$: La oss velge $n_0 = a$, slik at uttrykkene våre skal gjelde når $n \geq a$. Da er $0 \leq (n+a)^b \leq (n+n)^b \leq (2n)^b \leq 2^b n^b$, og $(n+a)^b \geq n^b$, så vi kan velge $c_1 = 1$ og $c_2 = 2^b$.

$a = 0$: Da er $(n+a)^b = n^b$; $c_1 = c_2 = n_0 = 1$ er én mulig løsning.

$a < 0$: La oss velge $n_0 = -2a$, slik at uttrykkene våre skal gjelde når $n \geq -2a$. Da er $a \geq -\frac{1}{2}n$, slik at $(n+a)^b \geq (n - \frac{1}{2}n)^b = (\frac{1}{2}n)^b = (\frac{1}{2})^b n^b$. Dessuten er $(n+a)^b \geq (-2a+a)^b = (-a)^b \geq 0$, og $(n+a)^b \leq (n+0)^b = n^b$. Altså kan vi velge $c_1 = (\frac{1}{2})^b$ og $c_2 = 1$.

Merk: Når man løser slike oppgaver trekker man naturligvis ikke tallene ut av løse luften, slik det ser ut når jeg skriver "La oss velge...". Først leker man seg med formelen og prøver å skrive den om, og så ser man hvilke forutsetninger man må gjøre for at omskrivningene skal gjelde.

3.1-3

Å si at en funksjon (f.eks. en som uttrykker kjøretiden til et program) er $O(n^2)$ betyr at funksjonen er medlem av mengden av funksjoner som har n^2 som øvre grense. Det er meningsløst å snakke om at en funksjon "minst" er medlem av en viss mengde.

3.1-4

Regnereglene for potenser gir følgende:

$$\begin{aligned} 2^{n+1} &= 2^n 2^1 = 2 \cdot 2^n \\ 2^{2n} &= (2^n)^2 \end{aligned}$$

Dermed er det enkelt å se at $2^{n+1} = 2 \cdot 2^n = O(2^n)$. Men $2^{2n} = (2^n)^2$ er jo faktisk kvadratet av 2^n , så 2^{2n} er ikke $O(2^n)$. Generelt gjelder følgende når a og b er konstanter (og $a \geq 0$):

$$\begin{aligned} a^{n+b} &= a^b a^n = \Theta(a^n) \\ a^{bn} &= (a^n)^b = \begin{cases} O(a^n) & \text{dersom } b \leq 1 \\ \Theta(a^n) & \text{dersom } b = 1 \\ \Omega(a^n) & \text{dersom } b \geq 1 \end{cases} \end{aligned}$$

3.1-5

La oss betrakte definisjonene av Θ , O og Ω :

$$\begin{aligned} \Theta(g(n)) &= \{f(n) : \text{det finnes positive konstanter } c_1, c_2 \text{ og } n_0 \text{ slik at} \\ &\quad 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \text{ for alle } n \geq n_0\} \\ O(g(n)) &= \{f(n) : \text{det finnes positive konstanter } c \text{ og } n_0 \text{ slik at} \\ &\quad 0 \leq f(n) \leq c g(n) \text{ for alle } n \geq n_0\} \\ \Omega(g(n)) &= \{f(n) : \text{det finnes positive konstanter } c \text{ og } n_0 \text{ slik at} \\ &\quad 0 \leq c g(n) \leq f(n) \text{ for alle } n \geq n_0\} \end{aligned}$$

La oss først tenke oss at $f(n) = \Theta(g(n))$. Da ser vi enkelt at ved å bruke c_2 som c i definisjonen av $O(g(n))$ og c_1 som c i definisjonen av $\Omega(g(n))$ og bruker samme n_0 begge steder er begge definisjonene oppfylt. La oss nå tenke oss at $f(n) = O(g(n))$ og $f(n) = \Omega(g(n))$. Dersom vi velger n_0 i definisjonen av $\Theta(g(n))$ lik den høyeste n_0 i de to andre definisjonene, c_1 lik c fra definisjonen av $\Omega(g(n))$ og c_2 lik c fra definisjonen av $O(g(n))$, ser vi at definisjonen av $\Theta(g(n))$ er oppfylt. Dermed har vi vist at $f(n) = \Theta(g(n)) \Leftrightarrow f(n) = O(g(n)) \wedge f(n) = \Omega(g(n))$.

3.1-6

Dersom kjøretiden er $O(g(n))$ i verste tilfelle og $\Omega(g(n))$ i beste tilfelle, er kjøretiden både $O(g(n))$ og $\Omega(g(n))$ uansett, siden kjøretiden selvfølgelig verken kan bli bedre enn i det beste tilfellet eller verre enn i det verste tilfellet. Dermed klarer vi oss med teorem 3.1, som vi beviste i forrige oppgave.

3.1-7

I følge definisjonen på side 48 inneholder $o(g(n))$ alle funksjoner $f(n)$ som er slik at uansett hvilken $c > 0$ vi velger, finnes det en $n_0 > 0$ slik at

$$0 \leq f(n) < c g(n)$$

for alle $n \geq n_0$. $\omega(g(n))$ er definert på tilsvarende måte, bortsett fra at her er det

$$0 \leq c g(n) < f(n)$$

som må gjelde. $o(n) \cap \omega(n)$ inneholder alle $f(n)$ som oppfyller *begge* kravene. Vi ser imidlertid at ingen slik $f(n)$ kan eksistere, siden de to kravene motsier hverandre. Begge kravene skal jo gjelde uansett verdien av c , så dersom vi velger samme c begge steder, kan jo ikke $f(n)$ både være større og mindre enn $c g(n)$ på samme tid.

Seksjon 3.2 - Treningsoppgaver

3.2-1

La $a > b$. Dersom $f(n)$ og $g(n)$ er monotont stigende, betyr det at $a > b \Rightarrow f(a) \geq f(b) \wedge g(a) \geq g(b)$. Vi ser da enkelt at

$$f(a) + g(a) \geq f(a) + g(b) \geq f(b) + g(b)$$

slik at $f(n) + g(n)$ er monotont stigende. Ettersom $g(a) \geq g(b)$ og $f(n)$ får større verdi jo større n er, gjelder også

$$f(g(a)) \geq f(g(b))$$

slik at $f(g(n))$ er monotont stigende. Til sist er det ganske greit å se at dersom både $f(n)$ og $g(n)$ er ikke-negative, er $f(n)g(n)$ monotont stigende også:

$$f(a) \cdot g(a) \geq f(a) \cdot g(b) \geq f(b) \cdot g(b)$$

Kapittel 3 - Oppgaver

3-2

A	B	O	o	Ω	ω	Θ	Merknader
$\lg^k n$	n^ϵ	ja	ja	nei	nei	nei	Dette bevises på side 54
n^k	c^n	ja	ja	nei	nei	nei	Eksponentialfunksjoner med grunntall større enn 1 tar <i>alltid</i> igjen polynomer, uansett grad
\sqrt{n}	$n^{\sin n}$	nei	nei	nei	nei	nei	Eksponenten i $n^{\sin n}$ oscillerer mellom -1 og 1
2^n	$2^{n/2}$	nei	nei	ja	ja	nei	$2^{n/2} = \sqrt{2}^n$
$n^{\lg c}$	$c^{\lg n}$	ja	nei	ja	nei	ja	I følge (3.15) på side 54 er disse to uttrykkene like
$\lg(n!)$	$\lg(n^n)$	ja	nei	ja	nei	ja	$\lg(n^n) = n \lg n$, og i følge (3.18) på side 55 er $\lg(n!) = \Theta(n \lg n)$

3-4

- Ikke korrekt. Moteksempel: $n = O(n^2)$, men $n^2 \neq O(n)$.
- Ikke korrekt. Moteksempel: $n + n^2 = \Theta(n^2) \neq \Theta(\min(n, n^2)) = \Theta(n)$.
- Korrekt. Bevis: Dersom $f(n) = O(g(n))$, er $0 \leq f(n) \leq cg(n)$ for alle n som er store nok. Da er

$$0 \leq \lg(f(n)) \leq \lg(CG(n)) = \lg c + \lg(g(n)).$$

Avhengig av størrelsen til c er $\lg c$ enten positiv, null eller negativ:

$$\lg c + \lg(g(n)) \leq \begin{cases} \lg(g(n)) & \text{dersom } \lg c < 0 \\ c_0 \lg(g(n)) & \text{dersom } \lg c \geq 0 \text{ (bare velg } c_0 \text{ og } n_0 \text{ store nok)} \end{cases}$$

slik at $\lg(f(n)) = O(\lg(g(n)))$.

- Ikke korrekt. Moteksempel: $2n = O(n)$, men $2^{2n} = (2^2)^n = 4^n \neq O(2^n)$.
- Ikke korrekt. Moteksempel: $\frac{1}{n} \neq O\left(\left(\frac{1}{n}\right)^2\right) = O\left(\frac{1}{n^2}\right)$
- Korrekt. Bevis: Dersom $f(n) = O(g(n))$, er $0 \leq f(n) \leq cg(n)$ for alle $n \geq n_0$. Da er $0 \leq \frac{1}{c}f(n) \leq g(n)$, slik at $g(n) = \Omega f(n)$.

7. Ikke korrekt. Moteksempel: $2^n \neq \Theta(2^{\frac{n}{2}}) = \Theta\left(\left(2^{\frac{1}{2}}\right)^n\right) = \Theta(\sqrt{2}^n)$.
8. Korrekt. Bevis: $o(f(n))$ er en mengde som inneholder alle funksjoner $h(n)$ som er slik at $0 \leq h(n) \leq cf(n)$ for alle $c > 0$ (bare n er stor nok). Ved å skrive $f(n) + o(f(n)) = \Theta(f(n))$ menes det at $f(n) + h(n) = \Theta(f(n))$, hvor $h(n)$ er plukket fra $o(f(n))$. Siden $f(n), h(n) \geq 0$, har vi

$$f(n) + h(n) \geq f(n) \geq 0.$$

Siden $h(n) \leq cf(n)$ for alle $c > 0$, gjelder

$$f(n) + h(n) \leq f(n) + cf(n) = (c + 1)f(n)$$

Dermed er $0 \leq f(n) \leq f(n) + h(n) \leq (c + 1)f(n)$, og siden $h(n)$ kunne være hva som helst fra $o(f(n))$, er $f(n) + o(f(n)) = \Theta(f(n))$.