DMA 2021
Ugeopgave 2

Aditya Fadhillah (hjg708)

17. november 2021

Del 1

Benyt definitionen af big-O fra denne uges noter til at bevise, at $n^2 + nlog_2(n)$ er $O(n^2)$.

$$f(x) \text{ er } O(g(x))$$

$$f(x) \leq c \cdot g(x)$$

$$n^2 + nlog_2(n) \leq c \cdot n^2$$

$$\Updownarrow$$

$$\frac{n^2}{n^2} + \frac{nlog_2(n)}{n^2} \leq c$$

$$\Updownarrow$$

$$1 + \frac{log_2(n)}{n} \leq c$$

Her ser man at c skal altid være større end 1. Og da $log_2(n)$ er udefineret når $n = 0$ skal $n_0 > 0$
For at bevise, at $n^2 + nlog_2(n)$ er $O(n^2)$, prøver jeg med $c = 2$ og $n_0 = 1$ i definitionen af store-O. Jeg tjekker for $f(x) \leq 2 * g(x)$, for $\forall n \geq 1$

$$n^2 + nlog_2(n) \leq 2 * n^2, \text{ for } \forall n \geq 1$$

$$\Updownarrow$$

$$0 \leq 2 \cdot n^2 - (n^2 + nlog_2(n)), \text{ for } \forall n \geq 1$$

$$\Updownarrow$$

$$0 \leq n^2 + n^2 - (n^2 + nlog_2(n)), \text{ for } \forall n \geq 1$$

$$\Updownarrow$$

$$0 \leq n^2 - (nlog_2(n)), \text{ for } \forall n \geq 1$$

Og da $0 \leq n^2 - (nlog_2(n))$, for $\forall n \geq 1$, kan vi se at n^2 er større end $nlog_2(n)$ for alle $n \geq 1$. Vi kan derfor konkludere at $f(x) \leq 2 \cdot g(x)$, for $\forall n \geq 1$

Del 2

Bevis følgende udsagn ved at bevise dets kontrapositive udsagn: Hvis r er et irrationelt tal, så er $r^{1/5}$ et irrationelt tal. Det er vigtigt at starte med at formulere det kontrapositive udsagn eksplicit!

Hvis r er et irrationelt tal, så er $r^{1/5}$ et irrationelt tal
Kontrapositive:
Hvis $r^{1/5}$ er rationelt tal, så er r et rationelt tal. Per definition af rationelt tal, består et rationelt tal af et heltal divideret med et andet heltal, altså $\frac{m}{n}$. Dvs. når $r^{\frac{1}{5}}$ er rationelt vil man kunne skrive det om til $r^{\frac{1}{5}} = \frac{m}{n}$. Og når $r^{\frac{1}{5}}$ er rationelt, så vil $(\frac{m}{n})^5 = r^{\frac{1}{5}} \cdot r^{\frac{1}{5}} \cdot r^{\frac{1}{5}} \cdot r^{\frac{1}{5}} \cdot r^{\frac{1}{5}} = r$ være rationelt. Ved at bevise kontrapositive af den oprindelig udsagn har jeg bevis udsagnet.

Del 3

Tag udgangspunkt i definitionen af store-O og bevis via modstrid, at 2^{2n} ikke er $O(2^n)$.

$$2^{2n} \text{ ikke er } O(2^n)$$

$$\text{Vi skal bevise via modstrid, Dvs.}$$

$$\text{Vi antager } 2^{2n} \text{ er } O(2^n)$$

$$\text{for } c > 0 \text{ og } n_0 > 0$$

$$f(x) \leq c \cdot g(x)$$

$$2^{2n} \leq c \cdot 2^n$$

$$\frac{2^n \cdot 2^n}{2^n} \leq \frac{c \cdot 2^n}{2^n}$$

$$2^n \leq c$$

Det er så ikke sandt da c er en konstant og 2^n er voksende, det betyder at på en større n vil 2^n på et tidspunkt blive større end c. Vi kan derfor konkludere at vi har bevis via modstrid, at 2^{2n} ikke er $O(2^n)$.

Del 4

For at løse et problem af størrelsen n, benytter algoritmerne A og B netop $T_A(n) = 2n^2log_2n$ og $T_B(n) = n^3$ mikrosekunder

Hvilken af de to algoritmer har den asymptotisk hurtigste køretid? Begrund dit svar med et formelt bevis, hvor du benytter asymptotisk notation. Du kan frit benytte teoremer og regler fra denne uges noter.

Fra regel (R5) ved vi at log_2n er $o(n)$, vi kan derefter bruge det til udregning af (R11) hvor $f(x) = log_2n$ og $g(x) = n$. (R11) siger (If $f(x)$ is $o(g(x))$ then $h(x)f(x)$ is $o(h(x)g(x))$.), så vi bruger $h(x) = 2n^2$. Det vil så betyde at $2n^2log_2n$ er $o(n^3)$.

Så det er algoritmen $T_A(n)$ som har den asymptotisk hurtigst køretid

Bestem en problemstørrelse ñ sådan, at det for alle n > ñ gælder, at algoritmen fra din besvarelse ovenfor er hurtigst.

Vi finder skæringspunktet mellem de to funktioner for at finde ñ.

$$T_A(n) = T_B(n)$$

$$2n^2log_2(n) = n^3$$

$$2n^2log_2(n) - n^3 = 0$$

$$n^2(2log_2(n) - n) = 0$$

Rødder:

$$n^2 = 0$$

$$2log_2(n) - n = 0$$

$$\Updownarrow$$

$$2log_2(n) - n + n = 0 + n$$

$$\Updownarrow$$

$$\frac{2log_2(n)}{2} = \frac{n}{2}$$

$$\Updownarrow$$

$$2^{log_2(n)} = 2^{\frac{n}{2}}$$

$$\Updownarrow$$

$$n = 2^{\frac{n}{2}}$$

Og det gælder for $n = 2$ og $n = 4$. Vi kan derfor bestemme problemstørrelse ñ for at være $n = 4$, sådan så at det for alle n > ñ gælder, at algoritmen $T_A(n)$ er hurtigst.