DM526: Supplerende note om

Asymptotisk notation

Torben Nielsen
tkn@imada.sdu.dk

8. oktober 2007

Baggrund

I analysen af algoritmer, interesserer vi os ofte for, hvordan køretiden og/eller pladsforbruget af en algoritme skalerer i forhold til inputtets størrelse. Typisk angives størrelsen af input som n – en konvention som vi også følger i denne note.

Beregner vi køretiden[1] af InsertionSort, ser vi at algoritmen bruger

$$\sum_{i=2}^{n}(i-1) = \frac{1}{2}n^2 - \frac{1}{2}n$$

sammenligninger i værste fald.

Dette ses af, at der i værste fald bruges én sammenligning for at behandle det andet element, to sammenligninger for at behandle det tredje element og så videre – op til $n-1$ sammenligning for den n'te element.

For at undgå komplicerede udtryk for køretid og pladsforbrug, "forkorter" vi ofte udtrykkene for disse til deres dominerende del. F.eks. siger vi, at InsertionSort har køretid ca. n^2, da udtrykket $\frac{1}{2}n^2 - \frac{1}{2}n$ *asymptotisk* opfører sig som n^2.

O og Ω

Som dataloger (eller matematikere) er vi naturligvis ikke tilfredse med "cirka". Vi definerer derfor følgende klasser af funktioner.

Definition 1. Lad $f(n)$ være en funktion. Da er mængden

$$O(f) = \{g \mid \text{der findes } c, n_0 > 0, \text{ så } 0 \le g(n) \le cf(n) \text{ for alle } n \ge n_0\}.$$

[1] Ved beregning af køretid, tæller man ofte antallet af instruktioner, der "tager lang tid". For sorteringsalgoritmer tæller man derfor ofteste antallet af sammenligninger.

Mængden $O(f)$ ("*store o af f*") er altså mængden af de funktioner, der asymptotisk vokser højst så hurtigt som f. Notationen O kan således bruges til at angive en asymptotisk øvre grænse. F.eks er det klart, at køretiden af InsertionSort er i $O(n^2)$ (tag f.eks. $c = 1$ og $n_0 = 1$).

Naturligvis findes der en tilsvarende notation til at angive asymptotiske nedre grænser.

Definition 2. Lad $f(n)$ være en funktion. Da er mængden

$$\Omega(f) = \{g \mid \text{der findes } c, n_0 > 0, \text{ så } 0 \le cf(n) \le g(n) \text{ for alle } n \ge n_0\}.$$

Mængden $\Omega(f)$ ("*store omega af f*") er altså mængde af de funktioner, der asymptotisk vokser mindst så hurtigt som f.

Som eksempel, beregner vi det mindste antal sammenligning InsertionSort skal udføre på en liste af n elementer. Uden bevis, påstår vi at InsertionSort bruger færrest sammenligning når input er sorteret. Da bruges der én sammenligning for hvert element i listen. Dvs. ialt n sammenligninger, hvilket klart ligger i $\Omega(n)$.

Θ, o og ω

De oftest brugte former for asymptotisk notation er formodentlig O og Ω, men det kan være nyttigt at have notation til at angive andre asymptotiske relationer.

Definition 3. Lad $f(n)$ være en funktion. Da er mængden

$$\Theta(f) = O(f) \cap \Omega(f).$$

Mængden $\Theta(f)$ ("*theta af f*") er altså mængde af de funktioner, der asymptotisk vokser præcis så hurtigt som f. Som eksempel kan nævnes at køretiden af SelectionSort er $\Theta(n^2)$ – algoritmen bruger altså i alle tilfælde et antal sammenligninger af størrelse n^2.

De sidste to asymptotisk relationer vi vil indføre notation for, er nok knapt så brugte som dem vi har set indtil videre – dog kan de være nyttige.

Definition 4. Lad $f(n)$ være en funktion. Da er mængden

$$o(f) = O(f) \setminus \Theta(f).$$

Mængden $o(f)$ ("*lille o af f*") er altså mængde af de funktioner, der asymptotisk vokser strengt langsommere end f.

Definition 5. Lad $f(n)$ være en funktion. Da er mængden

$$\omega(f) = \Omega(f) \setminus \Theta(f).$$

Mængden $\omega(f)$ ("*lille omega af f*") er altså mængde af de funktioner, der asymptotisk vokser strengt hurtigere end f.

Det er værd at bemærke at Θ, o og ω, alle kan defineres ved en "c og n_0"-definition i stil med O og Ω, skulle man have lyst til det.