Analyse af algoritmer

Eva Rotenberg*

Materialer Introduction to Algorithms, Cormen, Leiserson, Rivest og Stein (CLRS), kap. 3.

Opgaver

1 Relativ vækst

Arrangér følgende funktioner i voksende rækkefølge efter asymptotisk vækst, dvs., hvis funktionen $g(n)$ følger umiddelbart efter funktionen $f(n)$ i din liste skal der gælder at $f(n) = O(g(n))$.

$$n \log n \qquad n^2 \qquad 2^n \qquad n^3 \qquad \sqrt{n} \qquad n$$

2 Θ-notation.

Skriv følgende udtryk med Θ-notation

$$n^2 + \frac{n^3}{2}$$
$$2^n + n^4$$
$$\log_2 n + n\sqrt{n}$$
$$n(n-6)$$
$$4\sqrt{n}$$

$$8\log_2^7 n + 34\log_2 n + \frac{1}{1000}n$$
$$2^n \cdot 7 + 5\log_2^3 n$$
$$n(n^2 - 18)\log_2 n$$
$$n\log_2^4 n + n^2$$
$$n^3 \log_2 n + \sqrt{n}\log_2^7 n$$

3 Lykkelige løkker

Analysér køretiden af følgende løkker som funktion af n og udtryk resultatet i Θ-notation.

LØKKE1(n)
```
1: i = 1
2: while i ≤ n do
3:     print "∗"
4:     i = 2 · i
5: end while
```

LØKKE2(n)
```
1: i = 1
2: while i ≤ n do
3:     print "∗"
4:     i = 5 · i
5: end while
```

LØKKE3(n)
```
1: for i = 1 to n do
2:     j = 1
3:     while j ≤ n do
4:         print "∗"
5:         j = 2 · j
6:     end while
7: end for
```

* LØKKE4(n)
```
1: for i = 1 to n do
2:     j = i
3:     while j ≤ n do
4:         print "∗"
5:         j = 5 · j
6:     end while
7: end for
```

4 Asymptotiske påstande.

Hvilke af følgende påstande er korrekte?

$$\frac{1}{20}n^2 + 100n^3 = O(n^2)$$
$$\log_2 n + n = O(n)$$
$$2^{2\log_2 n} = O(n^2)$$
$$n^3(n-1)/5 = \Theta(n^3)$$
$$\log_2^2 n + n = \Theta(n)$$

$$\frac{n^3}{1000} + n + 100 = \Omega(n^2)$$
$$2^n + n^2 = \Omega(n)$$
$$\log_4 n + \log_{16} n = \Theta(\log n)$$
$$n^{1/2} + n^2 = \Theta(n)$$
$$2^{\log_4 n} = \Theta(\sqrt{n})$$

*baseret på materiale af Billes&Gørtz

Husk: Vi anvender notationen $\log^k(n) = (\log n)^k$.

5 Fordoblingshypoteser. Løs følgende opgaver:

5.1 [w] Algoritme A kører i præcis $7n^3$ tid på input af størrelse n. Hvor meget langsommere kører algoritmen hvis du fordobler inputstørrelsen?

5.2 Algoritme B kører på input af størrelsen 1000, 2000, 3000, 4000 og 5000 i henholdsvis 5, 20, 45, 80, og 125 sekunder. Estimér hvor lang tid det vil tage at køre Algoritme B på et input af størrelse 6000. Hvad er køretiden af algoritmen udtrykt i Θ-notation?

5.3 Algoritme C kører 3 sekunder langsommere hver gang man fordobler størrelsen af input. Hvad er køretiden af algoritmen udtrykt i Θ-notation?

6 Asymptotiske egenskaber

Løs følgende opgaver:

6.1 Lad $f(n)$ og $g(n)$ være asysmptotisk ikke-negative funktioner. Vis at $\max(f(n), g(n))$ er $\Theta(f(n) + g(n))$.

6.2 Forklar hvorfor udsagnet "køretiden af algoritme A er mindst $O(n^2)$" ikke giver mening.

6.3 Er $2^{n+1} = O(2^n)$? Er $2^{2n} = O(2^n)$?

6.4 Vis at $\log_2(n!) = O(n \log n)$.

6.5 [$*$] Vis at $\log_2(n!) = \Omega(n \log n)$.

- Kombiner opg. 6.5. med opg. 6.4. og konkluder at $\log_2(n!) = \Theta(n \log n)$.

7 Skæve fletninger

Professor Fl. Ette foreslår følgende variant af flettesortering kaldet 3-flettesortering. 3-flettesortering fungerer præcis som flettesortering pånær at man deler tabellen i 3 trejdedele, som sorteres rekursivt og flettes sammen. Løs følgende opgaver.

7.1 Vis at man kan flette 3 arrays i lineær tid.

7.2 Analysér køretiden af 3-flettesortering.

7.3 [$*$] Generaliser algoritmen og analysen af 3-flettesortering til k-flettesortering for $k > 3$. Er k-flettesortering en forbedring af standard 2-flettesortering?

8 Maksimalt delarray

Lad $A[0..n-1]$ være et array af heltal (både positive og negative). Et *maksimalt delarray* af A er et delarray $A[i..j]$ således at summen $A[i] + A[i+1] + \cdots + A[j]$ er maksimal blandt alle delarrays af A. Løs følgende opgaver:

8.1 [w] Giv en algoritme der finder et maksimalt delarray i A i $O(n^3)$ tid.

8.2 [†] Giv en algoritme der finder et maksimalt delarray i A i $O(n^2)$ tid. Hint: Vis at man kan beregne summerne for alle delarrays i $O(1)$ tid per delarray.

8.3 [$*$†] Giv en del-og-hersk algoritme der finder et maksimalt delarray i A i $O(n \log n)$ tid.

8.4 [$**$] Giv en algoritme, der finder et maksimalt delarray i A i $O(n)$ tid.