

Kjøretidsanalyse av rekursive algoritmer

Simon Jonassen, TDT4120, Høst 2005

22. september 2005

Innhold

1	Introduksjon	1
2	Iterasjonsmetoder	3
2.1	Foroversubstitusjon	3
2.2	Tilbakesubstitusjon	5
2.2.1	Variabelskifte	7
3	Substitusjon	9
4	Tremetoden	12
5	Masterteoremet	14
6	Litt om andre metoder	16
6.1	Empiriske metoder	16
6.2	Andre metoder	16

1 Introduksjon

Hovedpoenget er å måle kjøretid til rekursive algoritmer.

En algoritme er rekursiv dersom den kaller seg selv direkte eller indirekte (og evt. med en annen probleminstans) en eller flere ganger.

Framgangsmåte:

- velg en eller flere parametere som indikerer inputstørrelse (n)
- bestem algoritmens grunnleggende operasjon som en funksjon av inputstørrelse. ($1, n^k, n^k \log_j(n)$, enl.)
her kan man også skille mellom best-, average og worst case'r.
- sett opp en rekurrensligning og oppgi en initialverdi ($T(n) = \dots T(\dots) + \dots + f(n), T(1) = \dots$)
- løs rekurrensen på en eller en annen måte

2 Iterasjonsmetoder

2.1 Foroversubstitusjon

- Går fra $T(1)$ til $T(n)$
- Ser på $T(1), T(2), T(3), \dots$ inntil vi ser sammenhengen.
- Ganske enkel metode - som regel er det vanskelig å se løsning.
- Effektiv hvis en er flink i rekkeutviklinger

Et eksempel: Tower of Hanoi

$$T(n) = 2T(n - 1) + 1 \text{ for } n > 1, T(1) = 1$$

Løsning:

$$T(1) = 1$$

$$T(2) = 2 * T(1) + 1 = 2 + 1 = 3$$

$$T(3) = 2 * T(2) + 1 = 6 + 1 = 7$$

$$T(4) = 2 * T(3) + 1 = 14 + 1 = 15$$

Ser mønsteret nå: 1, 3, 7, 15, 31, 63, 127, 511, 1023, 2047, ...

$$T(n) = 2^n - 1 = \Theta(2^n)$$

2.2 Tilbakesubstitusjon

- Går fra $T(n)$ til $T(1)$
- Prøver å uttrykke $T(n)$ ved hjelp av $T(\text{noe som er mindre enn } n)$. Fortsetter iterativt inntil vi kommer til $T(1)$
- Poenget er å se mønsteret etter så få steg som bare mulig
- Metoden er relativt enkel (som regel)
- Effektiv hvis en er flink i rekkeutviklinger ;)

Eksempel:

$$T(n) = 2T(n-1) + 1 \text{ for } n > 1, T(1) = 1$$

Løsning:

$$T(n) = 2 * T(n-1) + 1$$

$$T(n) = 4 * T(n-2) + 2 + 1$$

$$T(n) = 8 * T(n-3) + 4 + 2 + 1$$

$$T(n) = 2^j * T(n-j) + 2^{j-1} + \dots + 2^2 + 2^1 + 2^0$$

$$T(n) = 2^{n-1} * T(n-n+1) + 2^{n-2} + \dots + 2^2 + 2^1 + 2^0$$

$$T(n) = \sum_{j=1}^n 2^{j-1}$$

$$\text{Rottmann: } T(n) = \frac{1-2^n}{-1} = 2^n - 1 = \Theta(2^n)$$

2.2.1 Variabelskifte

Vi kan erstatte n med noe som kan gi lettere løsning.

$$T(n) = 2 * T(n/2) + 1, T(1) = 1. \text{ Forslag?}$$

Løsning:

$$T(n) = 2 * T(n/2) + 1$$

la oss definere $2^m = n$, i så fall:

$$T(2^m) = 2 * T(2^{m-1}) + 1$$

la oss definere en $S(m)$ slik at $T(2^m) = S(m)$

$$S(m) = 2 * S(m - 1) + 1$$

Hanoi? $S(m) = 2^m - 1$

$$\text{Siden } T(n) = S(m) = 2^m - 1 = n - 1 = \Theta(n)$$

$$T(n) = 2 * T(\sqrt{n}) + 1, T(1) = 1$$

Løsning:

$$T(n) = 2 * T(\sqrt{n}) + 1$$

la oss definere $2^m = n$, i så fall

$$T(2^m) = 2 * T(2^{m/2}) + 1$$

samme trikset som før:

$$S(m) = 2 * S(m/2) + 1$$

dette har vi løst nettopp: $S(m) = m - 1$

$$T(n) = S(m) = \log_2(n) - 1 = \Theta(\log(n))$$

3 Substitusjon

Strategi: vi gjetter på en løsning for en enklere instans av T så forsøker å bevise at løsningen er gyldig og bestemmer konstanter.

Induksjonsbevis består av to steg:

- base: vis at regel gjelder for en instans $M(1)$
- induksjon: vis $M(j) \rightarrow M(j + 1)$

Et eksempel:

$$T(n) = T(n/2) + 1, T(1) = 1$$

Løsning:

$$T(n) = T(n/2) + 1 \text{ la oss gjette } T(n) \leq \log_2(n) \text{ for } n/2.$$

I så fall kan vi skrive at:

$$T(n) = T(n/2) + 1 \leq \log_2(n/2) + 1 = \log_2(n) - \log_2(2) + 1 = \log_2(n)$$

Ofte kan man bruke noe mer komplisert - da kommer det polynomer.

Vi kan ofte se bort fra konstante ledd i $T()$, f.eks. $T(n/3 + const)$

Vær forsiktig!

Feileksempel: $T(n) = 4T(n/2) + n$, antagelse er $T(n) \leq cn^2$

$$T(n) = 4T(n/2) + n \leq 4c(n/2)^2 + n = cn^2 + n = O(n^2) \Rightarrow \text{FEIL!}$$

Men løsningen er jo $O(n^2)$! Så hvordan skal man bevise det?

jo! siden vi har fått $cn^2 + n$ mens vi skulle få cn^2 kan vi bare trekke det siste leddet fra antagelsen våres, altså:

Anta at $T(n) \leq cn^2 - kn$ holder for $n/2$:

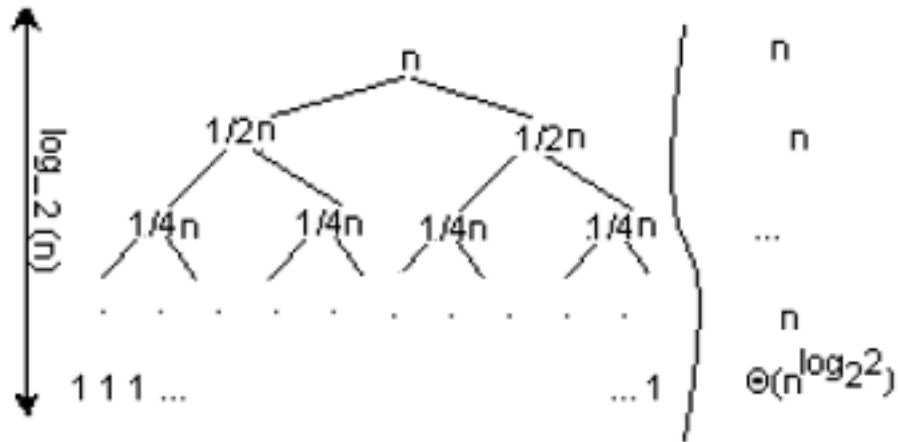
$$T(n) = 4T(n/2) + n = 4c((n/2)^2 - k(n/2)) + n = cn^2 - kn - (kn - n)$$

$$T(n) \leq cn^2 - kn, \text{ hvis } k \geq 1$$

4 Tremetoden

- Metoden er ganske intuitiv.
- Egner seg godt til å finne gode forslag

Eksempel: $T(n) = 2T(n/2) + n$



Rekursjonstre gir $T(n) = n * \log_2(n) = \Theta(n \log(n))$

5 Masterteoremet

La $a \geq 0$ og $b > 0$ være konstanter, la $f(n)$ være en funksjon, og la $T(n)$ være definert for ikke negative heltall med rekurensen $T(n) = aT(n/b) + f(n)$, hvor n/b betyr enten $\lfloor n/b \rfloor$ eller $\lceil n/b \rceil$.

- Hvis $f(n) = O(n^{\log_b(a)-\epsilon})$ for en $\epsilon > 0$ så $T(n) = \Theta(n^{\log_b(a)})$
- Hvis $f(n) = \Theta(n^{\log_b(a)})$ så $T(n) = \Theta(n^{\log_b(a)} \log(n))$
- Hvis $f(n) = \Omega(n^{\log_b(a)+\epsilon})$ for en $\epsilon > 0$, og hvis $af(n/b) \leq cf(n)$ for en konstant $c < 1$ for alle store n , så $T(n) = \Theta(f(n))$

Metoden er ganske lett, men det er en del småtteri en må være forsiktig med. T.d. logaritmer, regularitetskondisjon, ol.

Legg merke til at hvis det er $f(n) = \Theta(n^{\log_b(a)} \log^k(n))$ så $T(n) = \Theta(n^{\log_b(a)} \log^{k+1}(n))$ (aka en hemmelig versjon av case 2 ;))

Eksempler: $T(n) = 2T(n/2)+1$, $T(n) = 2T(n/2)+n$, $T(n) = 2T(n/2)+n \log(n)$, $T(n) = 2T(n/2) + n^2$

Løsning:

- $\Theta(n)$,case 1.
- $\Theta(n \log(n))$,case 2
- Ingen løsning med mt siden $n^\epsilon = \Omega(\log(n))$. (men f.eks. $T(n) = 3T(n/2) + n \log(n)$ skulle falle inn for case 1. Vær forsiktig)
- $\Theta(n^2)$, case 3

6 Litt om andre metoder

6.1 Empiriske metoder

Det er mulig å bruke inter- og ekstrapolasjon her. Men det finnes en liten juksemetode i tillegg :

- mål kjøretid i antall av prosedyrekall eller ms ($M(n)$)
- gjett en mulig $g(n)$.
- Se på $M(2n)/M(n)$ for flere n (hjelper når det er $\log_2(n)$)
- se på $M(n)/g(n)$ for flere n . Er forholdet konstant?
Ja: Du er så flink ;)
Nei: Bedre forslag?

6.2 Andre metoder

- Det finnes en HAUG med fine metoder som er bare laaangt utenfor pensum!
- Flere generaliserte metoder for løsning av lineære homogene og inhomogene rekursjonsrelasjoner med konstante koeffisienter:

- Kenneth H. Rosen «Discrete Mathematics and it's Applications» Kapittel 6.2 i 5. utgave