

SI-möte #5, Programkonstruktion och datastrukturer

Elias Castegren & Kristiina Ausmees

elca7381@student.uu.se || krau6498@student.uu.se

Övningar

1.

Antag att du ska tvätta alla dina strumpor. Eftersom du är programmerare har du kommit på följande algoritm:

1. Flytta en strumpa från tvättkorgen till tvättmaskinen.
2. Finns det strumpor kvar? Om ja, gå till steg 1, annars är du klar.
 - i)* Vad är det som avgör hur lång tid det tar att genomföra algoritmen, eller med andra ord: Vad avgör problemstorleken?
 - ii)* Ställ upp en rekursion som uttrycker hur antalet strumpförflyttningar beror av problemstorleken. Hur ser basfallet ut? Vad är problemstorleken för det mindre problemet (rekursionen) som dyker upp i det allmänna fallet?
 - iii)* Hitta en sluten formel för rekursionen, alltså en formel för hur många strumpförflyttningar man måste utföra om man har n strumpor.
 - iv)* Hur många fler förflyttningar måste man utföra om man dubblar antalet strumpor?

2.

Antag att du har tvättat och ska sortera dina strumpor som nu ligger i en hög. Varje par är unikt (en strumpa har exakt en annan strumpa som den hör ihop med) men eftersom du bara har svarta strumpor är det enda sättet att hitta ett passande par att jämföra en strumpa med alla andra strumpor i högen en och en tills du hittar rätt. Gör en värsta fall-analys:

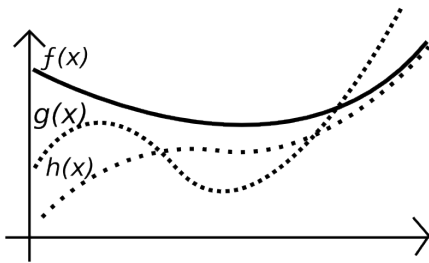
- i)* Skriv om ovanstående "algoritm" som en steg för steg-algoritm.
- ii)* Vad avgör problemstorleken?
- iii)* Hur många strumpjämförelser måste utföras för en strumpa i värsta fall?
- iv)* Ställ upp en rekursion som uttrycker hur antalet strumpjämförelser beror av problemstorleken. Hur ser basfallet ut? Vad är problemstorleken för rekursionen som dyker upp i det allmänna fallet?
- v)* Hur många strumpjämförelser måste man göra (i värsta fall) om man börjar med två, fyra, åtta eller 16 strumpor?
- vi)* Hitta en sluten formel för rekursionen, alltså en formel för hur många strumpjämförelser man måste utföra om man har n strumpor.
- vii)* Hur många fler jämförelser måste man utföra om man dubblar antalet strumpor?

Begrepp

- i) Vad innebär det matematiskt att $f(n) = O(g(n))$ eller att $y(x) = \Omega(u(x))$?
- ii) Vad betyder det att $f(x) = \Theta(g(x))$? Hur kan man uttrycka samma sak med hjälp av bara O och Ω ?
- iii) Vad innebär det att prata om en algoritms komplexitet? Vad är komplexiteten ett mått på? Vad är komplexiteten hos de två algoritmerna på föregående sida?
- iv) Vad kan man säga om två algoritmers körtid om man vet att den ena har högre komplexitet än den andra?

Fler övningar

3. Vilka av nedanstående påståenden är sanna (antag att graferna inte svänger mer än vad som visas):



- i) $g(x) = O(f(x))$
- ii) $f(x) = \Theta(h(x))$
- iii) $g(x) = \Theta(h(x))$
- iv) $h(x) = \Omega(f(x))$
- v) $h(x) = O(f(x))$

4. Fyll i de blanka fälten så att nedanstående samband stämmer (ange så enkla funktioner som möjligt). Använd de formella definitionerna av Θ , Ω och O om du är osäker och vill bevisa något samband. Du kan anta att alla n är heltal större än 0.

- i) $3n^3 + 15n^2 + 3 = \Theta(\dots)$
- ii) $\dots = O(42)$
- iii) $n \cdot \log_{10} n + 3 = \Omega(\dots)$
- iv) $n^2 + 100n + 15 = O(\dots)$
- v) $n \cdot \log_2 n + n = \Theta(\dots)$
- vi) $n^2 + \log_2 n = \dots (n \cdot \log_2 n)$
- vii) $2^n = \dots (n!)$

5. Betrakta nedanstående funktioner:

```
fun f(0) = 0
| f(n) = if 2*n>42 then 2*f(n-1) else f(n-1)+1;
```

```
fun g(0) = 1
| g(n) = g(n-1) + g(n-1);
```

Ge en rekursiv beskrivning av tidskomplexiteten $T(n)$ och finn en sluten formel för varje rekursion, t.ex. genom att gissa en sluten formel och bevisa den med induktion. Bestäm tidskomplexiteten med asymptotiskt närliggande gränser (Θ).

Lycka till!