

LÖSNINGAR algoritmik, 17 okt 1997

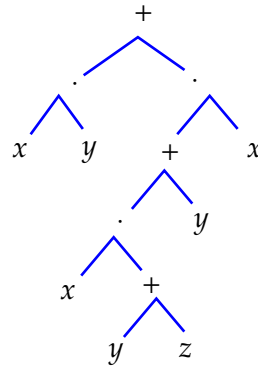
- 1 Låt w vara nedanstående aritmetiska uttryck

$$((x \cdot y) + ((x \cdot (y + z)) + y) \cdot x))$$

LÖSNING

- a) Rita ett binärt träd för w .

a)



- b) Räkna upp det binära trädets noder i preordning.

b) $+ \cdot x y \cdot + \cdot x + y z y x$

- c) Presentera en primitivt rekursiv funktionsomberäknar w .

c) $Add(Mult(x, y),$
 $Mult(Add(Mult(x,$
 $Add(y, z)),$
 $y),$
 $x))$

- 2 Betrakta proceduren

```

r ← 1
Repetera med k = 0 till x{
  i ← k
  Öka i
  Addera i och k i m
  Multiplicera r med m
  n ← m
  Öka n
  Multiplicera r med n}
    
```

LÖSNING

- a) Vilket innehåll får r om proceduren körs med $x = 2$, $x = 3$ respektive med godtyckligt x ?

a) $x = 2$ ger r innehållet $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6$, $x = 3$ ger r innehållet $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8$, och för godtyckligt x får r innehållet $1 \cdot 2 \cdot \dots \cdot n$ där $n = 2(x + 1)$.

- b) Konstruera en primitivt rekursiv funktion som beräknar samma sak som proceduren.

b) $f(x) = g(Mult(2, Öka(x)))$ där $\begin{cases} g(0) = 1 \\ g(Öka(x)) = Mult(Öka(x), g(x)) \end{cases}$

3 Funktionen f definieras med hjälp av funktionen h .

Bestäm $f(0)$, $f(1)$, $f(2)$, $f(7)$, $f(8)$ och $f(x)$ för godtyckligt x .

Visa hur Du gjorde.

$$f(x) = h(x, 0)$$

$$h(x, y) = \text{Om}(\text{Icke}(\text{Mindre}(\text{Pot}(2, y), x)), \\ y, \\ h(x, \text{Öka}(y)))$$

LÖSNING

$$\text{Pot}(2, 0) \geq 0$$

$$f(0) = h(0, 0) = 0$$

$$\text{Pot}(2, 0) \geq 1$$

$$f(1) = h(1, 0) = 0$$

$$\text{Pot}(2, 0) < 2 \quad \text{Pot}(2, 1) \geq 2$$

$$f(2) = h(2, 0) = h(2, 1) = 1$$

$$\begin{array}{ccccccc} & & & & \text{Pot}(2, 3) \geq 7 & & \\ & & & \text{Pot}(2, 2) < 7 & & & \\ & \text{Pot}(2, 1) < 7 & & & & & \\ \text{Pot}(2, 0) < 7 & & & & & & \\ f(7) = h(7, 0) = h(7, 1) = h(7, 2) = h(7, 3) = 3 \end{array}$$

$$\begin{array}{ccccccc} & & & & \text{Pot}(2, 3) \geq 8 & & \\ & & & \text{Pot}(2, 2) < 8 & & & \\ & \text{Pot}(2, 1) < 8 & & & & & \\ \text{Pot}(2, 0) < 8 & & & & & & \\ f(8) = h(8, 0) = h(8, 1) = h(8, 2) = h(8, 3) = 3 \end{array}$$

För godtyckligt x är $f(x)$ lika med det minsta heltal y , sådant att $2^y \geq x$. Ty y rekurserar framåt ett steg i taget så länge $2^y < x$. Och vid först påträffade y sådant att $2^y \geq x$ upphör rekursionen och nämnda y returneras.

4 Vad beräknar funktionen $Våg$ vid körning på binära träd?

$$\begin{cases} Våg([]) = 1 \\ Våg([rot \ v\ddot{a} \ h\ddot{o}]) = Lika(g(v\ddot{a}), g(h\ddot{o})) \\ \\ g([]) = 0 \\ g([rot \ v\ddot{a} \ h\ddot{o}]) = \ddot{O}ka(Add(g(v\ddot{a}), g(h\ddot{o}))) \end{cases}$$

LÖSNING

Funktionen g beräknar (returnerar) antalet noder i det givna binära trädet. Funktionen $Våg$ reder ut huruvida det givna binära trädet har lika många noder i sitt vänstra underträd som i sitt högra, dvs $Våg$ returnerar 1 om trädets har lika många noder i vänster underträd som i höger, och 0 annars.

5 Konstruera en listfunktion som för en (enkel) tallista L reder ut om det finns någon nod i listan som innehåller

a) listans längd,

b) nodens nummer, där listans första nod, andra nod, osv ... har nummer 1, 2, osv

LÖSNING

$$\text{a) } \text{Inneh\ddot{a}llerSinL\ddot{a}ngd?}(L) = h(L, \text{L\ddot{a}ngd}(L))$$

$$\begin{cases} h([], y) = 0 \\ h([nod|L], y) = \text{Eller}(\text{Lika}(nod, y), h(L, y)) \end{cases}$$

KOMMENTAR: Notera att längden för den givna listan skickas med som (andra) argument till hjälpfunktionen h . Under rekursionen i h 's första argument jämförs den medskickade längden med den givna listans noder.

$$\text{b) } \text{Inneh\ddot{a}llerN\ddot{a}gotNodnummer?}(L) = h(L, 1)$$

$$\begin{cases} h([], n) = 0 \\ h([nod|L], n) = \text{Eller}(\text{Lika}(nod, n), h(L, \text{Öka}(n))) \end{cases}$$

6 Visa hur nummerfunktionen nedanför kan beräknas inom klassen av primitivt rekursiva funktioner.

Funktionen *Nummer* räknar upp talparen i den ordning som de träffas på när man följer pilarnas väg i figuren.

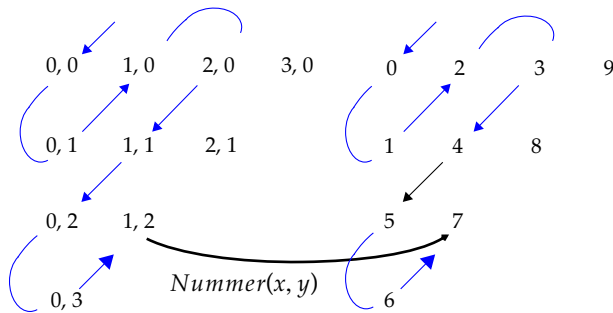
Dvs. så att

$Nummer(0, 0) = 0, Nummer(0, 1) = 1,$

$Nummer(1, 0) = 2, Nummer(2, 0) = 3,$ osv

...

par av naturliga tal och deras nummer



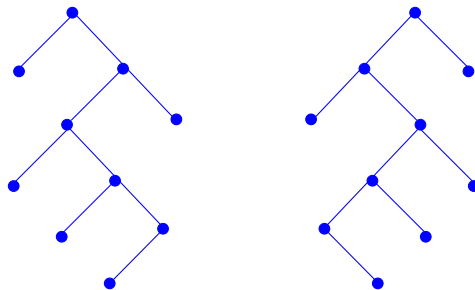
9

LÖSNING

Talparen och deras nummer beskrivs i den pilmarkerade vägen, vilken går längs diagonaler. Om man följer den pilmarkerade vägen så långt att man just har gått igenom en full triangel av diagonaler är antalet genomgångna talpar lika med ett triangeltal, närmare bestämt det $(x + y)$:te triangeltalet, om x, y är ett talpar på nästkommande diagonal. I detta läge står man i begrepp att påbörja en ny diagonal uppifrån eller nerifrån. Det förra inträffar om nästkommande talpar befinner sig på en diagonal sådan att summan av de två talen är jämn. Det senare om samma summa är udda. Notera också att y -talet i talparen börjar med 0 och ökar i takt med att man följer en nedåtgående diagonal, medan x -talet ändras på samma sätt när man följer en uppåtgående diagonal. Härav,

$$Nummer(x, y) = Om(Jämn(Add(x, y)), \\ Add(y, Triangeltal(Add(x, y))), \\ Add(x, Triangeltal(Add(x, y))))$$

7 Skriv en funktion som för två godtyckliga binära träd reder ut om de är spegelbilder av varandra. Tag hänsyn enbart till själva trädstrukturerna, inte till nodinnehåll.



LÖSNING

Ett binärt träd är spegelbild av ett annat, om det förras vänstra underträd är spegelbild av det senares högra underträd och det förras högra underträd är spegelbild av det senares vänstra underträd. Som läsaren säkert har noterat är denna beskrivning av spegelbilder en rekursiv beskrivning. Återstår basfall. De minsta binära träden – de tomma – är onekligen spegelbilder av varandra. Och ett tomt binärt träd är inte spegelbild av ett icke tomt. Härav:

$$\begin{cases} \text{Spegel}([], []) = 1 \\ \text{Spegel}([], [\text{rot vä hö}]) = 0 \\ \text{Spegel}([\text{rot vä hö}], []) = 0 \\ \text{Spegel}([\text{rot vä hö}], [\text{rot' vä' hö'}]) = Och(\text{Spegel}(\text{vä}, \text{hö}), \\ \text{Spegel}(\text{hö}, \text{vä'})) \end{cases}$$