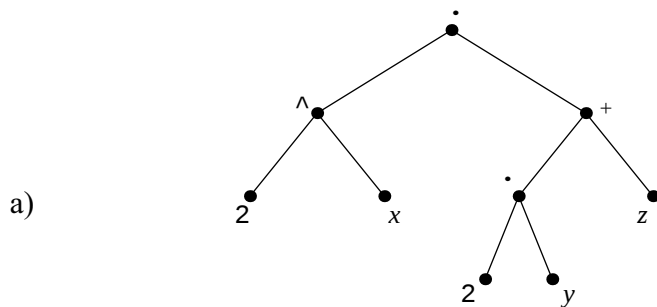


## Kortfattade lösningar ALGORITMIK SVP, 15 dec 99

1

a) Rita det binära träd vars noder i preordning är  $\cdot \wedge 2 x + \cdot 2 y z$  (2p)

b) Räkna upp trädets noder i post- och inordning. (2p)



b) Postordning:  $2 x \wedge 2 y \cdot z + \cdot$  inordning:  $2 \wedge x \cdot 2 \cdot y + z$

2 Vilket innehåll kommer högen  $r$  att få om man kör proceduren OjOj med  $x = 1, 2, 3, 4$ . (Redovisa Dina provkörningar.) Försök också lista ut vilket innehåll som  $r$  får för godtyckligt  $x$ ? (5p)

OjOj  $x$  i  $r =$

$y \leftarrow 1$

$z \leftarrow 1$

$r \leftarrow 0$

Så länge  $z \leq x$  {

Öka  $r$

Öka  $y$

Öka  $y$

Addera  $y$  till  $z$

}

|     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| $x$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| $r$ | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 3 | 3  | 3  | 3  | 3  | 3  | 3  | 4  | 4  |

Som synes, innehållet i  $r$  blir heltalsdelen av  $\sqrt{x}$ .

3

$\begin{cases} f(0) = 1 \\ f(\text{Öka}(x)) = \text{Mult}(f(x), \text{Om}(\text{Prima?}(x), x, 1)) \end{cases}$

Bestäm  $f(3)$ ,  $f(4)$ ,  $f(5)$ ,  $f(6)$ .

(4p)

|        |   |   |   |   |             |             |                     |
|--------|---|---|---|---|-------------|-------------|---------------------|
| $x$    | 0 | 1 | 2 | 3 | 4           | 5           | 6                   |
| $f(x)$ | 1 | 1 | 1 | 2 | $2 \cdot 3$ | $2 \cdot 3$ | $2 \cdot 3 \cdot 5$ |

4 Givet ett godtyckligt träd  $T$  (inte binärt), konstruera en funktion som returnerar antalet noder som inte är löv i  $T$ . (5p)

$$\text{AntaletFöräldrar}([ ]) = 0$$

$$\begin{aligned} \text{AntaletFöräldrar}(T) = & \text{Add}(\text{Om}(\text{Lika}(1, \text{Längd}(\text{UtomSista}(T))), \\ & 1, \\ & \text{AntaletFöräldrar}(\text{UtomSista}(T))), \\ & \text{AntaletFöräldrar}(\text{Sista}(T))) \end{aligned}$$

5 Antag att  $\text{Låg}(L)$  returnerar en lista innehållande de element ur en enkel tallista  $L$  som är lägre än  $L$ :s första element, och antag att  $\text{Hög}(L)$  returnerar de övriga elementen i  $L$  förutom  $L$ :s första element. Vad returnerar nedanstående funktion  $\text{Hipp}$  när den körs på listan  $[3 \ 5 \ 1 \ 1 \ 0 \ 3 \ 6 \ 7 \ 2]$ ? Och vad gör  $\text{Hipp}$  för godtycklig lista av tal? (5p)

$$\begin{cases} \text{Hipp}([ ]) = [ ] \\ \text{Hipp}([nod \mid L]) = \text{FogaSamman}(\text{Hipp}(\text{Låg}([nod \mid L])), \text{FogaIn}(nod, \text{Hipp}(\text{Hög}([nod \mid L]))) \end{cases}$$

$\text{Hipp}$  sorterar en lista från mindre till större. (Kallas vanligtvis *QuickSort*.)

$$\text{Hipp}([3 \ 5 \ 1 \ 1 \ 0 \ 3 \ 6 \ 7 \ 2]) = [0 \ 1 \ 1 \ 2 \ 3 \ 3 \ 5 \ 6 \ 7]$$

6 Konstruera listfunktionerna  $\text{Låg}$  och  $\text{Hög}$  från föregående uppgift. Visa sedan med provkörning hur de fungerar. (6p)

Låt oss slå två flugor i en smäll (med hjälp av funktionen  $h$ :

$$\text{Låg}([ ]) = [ ]$$

$$\text{Hög}([ ]) = [ ]$$

$$\text{Låg}([nod \mid L]) = \text{Första}(h(nod, L, [ ], [ ]))$$

$$\text{Hög}([nod \mid L]) = \text{Sista}(h(nod, L, [ ], [ ]))$$

$$h(nod, [ ], L_1, L_2) = [L_1 \ L_2]$$

$$\begin{aligned} h(nod, [x \mid L], L_1, L_2) = & \text{Om}(\text{Mindre}(x, nod), \\ & h(nod, L, \text{FogaIn}(x, L_1), L_2), \\ & h(nod, L, L_1, \text{FogaIn}(x, L_2))) \\ & ) \end{aligned}$$

7 Konstruera en funktion som för ett godtyckligt binärt träd  $T$  (som antas ha naturliga tal i sina noder) och ett godtyckligt naturligt tal  $x$  returnerar antalet förekomster av  $x$  i  $T$ 's noder (4p)

$$\begin{aligned} \text{AntalFörekomster}(x, []) &= 0 \\ \text{AntalFörekomster}(x, [\text{rot } T_{\text{vä}} \ T_{\text{hö}}]) &= \\ &\quad \text{Add}(\text{Lika}(x, \text{rot}), \\ &\quad \quad \text{Add}(\text{AntalFörekomster}(x, T_{\text{vä}}), \\ &\quad \quad \quad \text{AntalFörekomster}(x, T_{\text{hö}})) \\ &\quad ) \end{aligned}$$

8 Funktionen *Union* returnerar en lista av samtliga skilda element som finns i den ena eller andra (eller i båda) av två listor. (7p)

$$\begin{cases} \text{Union}([], Y) = Y \\ \text{Union}([nod \mid X], Y) = \text{Om}(\text{Tillhör}(nod, \text{Union}(X, Y)), \text{Union}(X, Y), [nod \mid \text{Union}(X, Y)]) \end{cases}$$

Din uppgift är att

- redovisa en provkörning av funktionen *Union* på listorna  $[a \ b \ a]$ ,  $[d \ b]$ .
- konstruera en funktion *Snitt* som returnerar en lista av samtliga skilda element som är gemensamma för två listor. Nedanför ges några illustrationer.

|                            |                                |
|----------------------------|--------------------------------|
| två listor                 | en lista av gemensamma element |
| $[a \ b \ c], [a \ d]$     | $[a]$                          |
| $[a \ b \ a], [a \ d \ b]$ | $[a \ b]$                      |

a) Man får ...  $[a \ d \ b]$

b)  $\text{Snitt}(X, Y) = h([], X, Y)$

$h(Z, [], Y) = Z$

$h(Z, [nod \mid X], Y) = \text{Om}(\text{Tillhör}(nod, Y), h(\text{Union}(Z, [nod]), X, Y), h(\text{Union}(Z, []), X, Y))$