– Arbori de tipul (2) –

(r () ())

subarb. st. ↙ ↘ subarb. dr

8. Să se construiască lista nodurilor unui arbore de tipul (2) parcurs în inordine.

Exemplu: (1 (2 (4 () (7)) ()) (3 (5) (6))) → (4 7 2 1 5 3 6)

Model recursiv

$$inordine(l_1\ l_2\ l_3) = \begin{cases} \emptyset, & \text{dacă } l \text{ e vidă} \\ inordine(l_2) \oplus l_1 \oplus inordine(l_3), & \text{altfel} \end{cases}$$

---

10. Se dă un arbore de tipul (2). Să se precizeze nivelul pe care apare un nod x în arbore. Nivelul este 0.

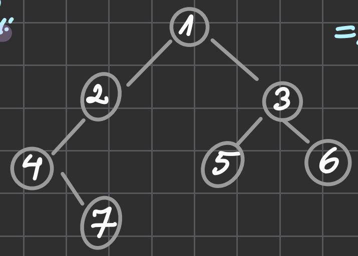
Model recursiv

$$nivel(l_1\ l_2\ l_3, x, nivel) = \begin{cases} nil, & \text{dacă } l = \emptyset \\ nivel, & \text{dacă } x = l_1 \\ nivel(l_2, x, nivel+1) \,||\, nivel(l_3, x, nivel+1), & \text{altfel} \end{cases}$$

---

11. Se dă un arbore de tipul (2). Să se afișeze nivelul (și lista corespunzătoare a nodurilor) având număr maxim de noduri. Nivelul răd. se consideră 0.

Exemplu:



=> nivelul 2 are cele mai multe noduri

(1 (2 (4 () (7)) ()) (3 (5) (6)))

Model recursiv

$$nr\_noduri(l, nivel, nivelc) = \begin{cases} 0, & \text{dacă } l = \emptyset \\ 1 + nr\_noduri(l_2, nivel, nivelc+1) + nr\_noduri(l_3, nivel, nivelc+1), & \text{dacă } nivel = nivelc \\ \text{altfel, nu adaug 1} \end{cases}$$

$$nr\_nivele(l, nivele) = \begin{cases} nivele, & \text{dacă } l = \emptyset \\ max(nr\_nivele(l_2, nivele+1), nr\_nivele(l_3, nivel+1)), & \text{altfel} \end{cases}$$

nivel_maxim(l, nrnivele, nivelCurent, maxim)

$$\begin{cases} maxim, & \text{dacă } nrNivele < nivelCurent \\ nivel\_maxim(l, nrNivele, nivelCurent+1, \overbrace{nr\_noduri(l, nivelCurent, 0)}^{*}), & \text{dacă } * > maxim \\ nivel\_maxim(l, nrNivele, nivelCurent+1, maxim), & \text{altfel} \end{cases}$$

listă_noduri(l, nivelc, nivelmaxim) =

$$\begin{cases} (), & \text{dacă } l = \emptyset \\ l_1 \oplus listă\_noduri(l_2, nivelc+1, nivelmaxim) \oplus listă\_noduri(l_3, nivelc+1, nivelmaxim), & \text{dacă } nivelc = nivelmaxim \\ listă\_noduri(l_2, nivelc+1, nivelmaxim) \oplus listă\_noduri(l_3, nivelc+1, nivelmaxim), & \text{altfel} \end{cases}$$

(12) Să se construiască lista nodurilor unui arbore de tipul (2) parcurs în preordine.

Model recursiv

$$preordine(l_1 l_2 l_3) = \begin{cases} \emptyset, & \text{dacă } l = \emptyset \\ l_1 \oplus preordine(l_2) \oplus preordine(l_3), & \text{altfel} \end{cases}$$

---

(13) Se dă un arbore de tipul (2). Să se afișeze calea de la rădăcină până la un nod x dat.

Model recursiv

$$există(l, x) = \begin{cases} \emptyset, & \text{dacă } l = \emptyset \\ x, & \text{dacă } l_1 \text{ e atom și } l_1 = x \\ există(l_2, x), & \text{dacă } l_1 \text{ e atom} \\ există(l_1, x) \,||\, există(l_2, x), & \text{dacă } l_1 \text{ e listă} \end{cases}$$

$$cale(l, x) = \begin{cases} \emptyset, & \text{dacă } l = \emptyset \\ (l_1), & \text{dacă } l_1 = x \\ l_1 \oplus cale(l_2, x), & \text{dacă } există(l_2, x) \\ l_1 \oplus cale(l_3, x), & \text{dacă } există(l_3, x) \\ nil, & \text{altfel} \end{cases}$$

---

(14) Să se construiască lista nodurilor unui arbore de tipul (2) parcurs în postordine. (SDR)

Model recursiv

$$postordine(l_1 l_2 l_3) = \begin{cases} \emptyset, & \text{dacă } l = \emptyset \\ postordine(l_2) \oplus postordine(l_3) \oplus l_1, & \text{altfel} \end{cases}$$

---

(16) Să se decidă dacă un arbore de tipul (2) este echilibrat (dif dintre adâncimile celor 2 subarbori nu este mai mare decât 1).

Model recursiv

$$nr\_nivele(l, nivele) = \begin{cases} nivele, & \text{dacă } l = \emptyset \\ max(nr\_nivele(l_2, nivele+1), nr\_nivele(l_3, nivele+1)) \end{cases}$$

$echilibrat(l) =$

$$\begin{cases} t, & \text{dacă } |nr\_nivele(l_2, 0) - nr\_nivele(l_3, 0)| = 1 \\ f, & \text{altfel} \end{cases}$$

=> NU e echilibrat

---

(4) Să se convertească un arbore de tipul (2) la un arbore de tipul (1).

Model recursiv

$$convertire(l) = \begin{cases} \emptyset, & \text{dacă } l = \emptyset \\ (l_1\ 2) \oplus convertire(l_2) \oplus convertire(l_3), & \text{dacă } l_2 \text{ și } l_3 \neq \emptyset \\ (l_1\ 1) \oplus convertire(l_2) \oplus convertire(l_3), & \text{dacă } l_2 \text{ sau } l_3 \neq \emptyset \\ (l_1\ 0) \oplus convertire(l_2) \oplus convertire(l_3), & \text{dacă } l_2 \text{ și } l_3 = \emptyset \end{cases}$$