

eljárás törlés (gyökér; szülő, p : mutatótípus);

változó gyerekek : mutatótípus;

ha elem [p].bal <> vége akkor

gyerekek := elem [p].bal

különben

gyerekek := elem [p].jobb;

vége;

ha p = gyökér akkor

gyökér := gyerekek

különben

ha p = elem [szülő].bal akkor

elem [szülő].bal := gyerekek

különben

elem [szülő].jobb := gyerekek;

vége;

vége;

vége;

eljárás töröl<sub>2</sub> (gyöker, szülő, p: mutatótípus);

változó töv, kövszülő: mutatótípus;

kövszülő := p;

töv := elem [p]. jobb;

amíg elem [p]. bal  $\neq$  végjel kiútkel

kövszülő := töv; töv := elem [p]. bal;

a vége;

ha elem [töv]. jobb = végjel akkor

töröl<sub>1</sub> (kövszülő, töv)

különben

töröl<sub>1</sub> (kövszülő, töv);

h vége;

ha p = gyöker akkor

gyöker := töv

különben

ha p = elem [szülő]. bal akkor

elem [szülő]. bal := töv

különben

elem [szülő]. jobb := töv;

h vége;

h vége;

elem [töv]. bal := elem [p]. bal;

elem [töv]. jobb := elem [p]. jobb;

vége;

függvény keresőfatöröl (gyökér: mutatótípus; Adat: értéktípus): logikai;

változó: szülő, p: mutatótípus;

szülő := végjel; p := gyökér;

amíg (p <> végjel) és (elem [p].érték <> Adat) ismételt

szülő := p;

ha Adat < elem [p].érték akkor

p := elem [p].bal

különben

p := elem [p].jobb;

li vége;

a vége;

ha p < végjel akkor

ha (elem [p].bal = végjel) és (elem [p].jobb = végjel) akkor

töröl  $\emptyset$  (szülő, p)

különben

ha (elem [p].bal = végjel) vagy (elem [p].jobb = végjel) akkor

töröl 1 (szülő, p)

különben

töröl 2 (szülő, p)

li vége;

li vége;

felismerés(p);

keresőfatöröl := igaz

különben

keresőfatöröl := hamis;

li vége;

vége;

## VISSZALÉPÉSES KERESÉS:

(8 irányú probléma megoldása)

deklarálás:

konstans  $n = 8$ ;

változó  $x$ : tömb  $[1..n]$  egész;

Az algoritmus(ok):

függvény visszalépes: logikai;

változó  $i$ : egész;

$i := 1$ ;  $x[i] := \emptyset$ ;

amíg  $(i > 1)$  és  $(i \leq n)$  ismét

ha találat helyett  $(i)$  akkor

$i := i + 1$ ;

ha  $i \leq n$  akkor

$x[i] := \emptyset$ ;

h vége

különben

$i := i - 1$ ;

h vége;

a vége;

visszalépeses :=  $i > n$ ;

vége;

függvény találgelyet ( $i$ : egész): logikai;

$x[i] := x[i] + 1$ ;

amíg  $(x[i] \leq w)$  és van\_ütes( $i$ ) ismét

$x[i] := x[i] + 1$ ;

a vége;

találgelyet :=  $x[i] \leq w$ ;

vége;

függvény van\_ütes( $i$ : egész): logikai;

változó  $j$ : egész;

$j := 1$ ;

amíg  $(j < i)$  és  $(x[i] < x[j])$  és

$(abs(x[i] - x[j]) < i - j)$  ismét

$j := j + 1$ ;

a vége;

van\_ütes :=  $j < i$ ;

vége;

eljárás megoldás;

változó  $i$ : egész;

célus  $i := 1..w$  ismét

$k_i := x[i]$

c vége;

vége;

8 irányú probléma összes megoldása:

konstans  $n = 8$ ;

változó  $x$ : tömb  $[1..n]$  egész;

változó  $D_k, k$ : egész,

"  $D_k := \emptyset$ ;  $\varepsilon := 1$ ;

$x \in \mathbb{Z} := \emptyset$ ;

amíg  $k \geq 1$  ismét

ha visszalépés ( $k$ ) akkor

$D_k := D_{k+1}$ ;

$\varepsilon_i := D_k$ , "megoldás"

megoldásk

$\varepsilon := \varepsilon - 1$ .

különb

$\varepsilon_i$ : "Nincs (több) megoldás"

k vége;

a vége;

függvény visszalépései ( $i$ : egész): logikai;

amíg  $(i \geq 1)$  és  $(i \leq u)$  igaz

ha. találat (i) akkor

$i := i + 1;$

ha  $i \leq u$  akkor

$x[i] := \emptyset;$

h. vége

különben

$i := i - 1;$

h. vége;

a. vége;

visszalépések:  $= i > u;$

vége;

Zárójel ellenőrzése veremmel:

be: kifejezés;

veremből (verem);

$i := 1$ ;  $oE1 := igaz$ ;  $oE2 := igaz$ ;

amíg ( $i \leq \text{hossz}(\text{kifejezés})$ ) és  $oE1$  és  $oE2$  igaz

ha kifejezés  $[i] = "("$  akkor

$oE1 := \text{veremből}(\text{verem}, \text{kifejezés}[i])$ ;

h vége;

ha kifejezés  $[i] = ")"$  akkor

$oE2 := \text{veremből}(\text{verem}, \text{jel})$ ;

h vége;

$i := i + 1$ ;

a vége;

ha nem  $oE1$  akkor

ki: "túlcsordult a verem!"

különben

ha nem  $oE2$  akkor

ki: "(" hiányzik"

különben

ha nem  $\text{veremüres}(\text{verem})$  akkor

ki: ")" hiányzik"

különben

ki: "konkt"

h vége;

h vége;

h vége;