

Törles a lista végén:

függvény listatörölveg (fej: mutatótípus): logikai;

változó pe, p: mutatótípus;

ha fej <> végjel akkor

ha elem [fej].kör = végjel akkor

listatörölveg := listatörölveg (fej).

különben

pe := fej; p := elem [fej].kör;

amíg elem [p].kör <> végjel ismételt

pe := p; p := elem [p].kör;

a vége;

listatörölveg := listatörölmögül (pe);

h vége;

különben

listatörölveg := hamis;

h vége;

vége;

11. Rendezett lista:

Beszűrés rendezett listába:

függvény listafűrésd (f_{ej} : mutatótípus; Adat: értékípus):
logikai;

változó pe, p : mutatótípus;

ha ($f_{ej} = végjel$) vagy ($Adat < elem [f_{ej}] . érték$) akkor
listafűrésd := listafűréselő (f_{ej} , Adat)

különben

$pe := f_{ej}$; $p := elem [f_{ej}] . érték$;

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

$pe := p$; $p := elem [p] . érték$;

a végc;

listafűrésd := listafűréselő (pe , Adat);

a végc;

végc;

Törés rendezett listából:

függvény listatörrend (f_{ij} : mutatótípus; Adat: értékpus): logikai;

változó pe, p : mutatótípus;

ha $f_{ij} \leftrightarrow végjel$ akkor

ha $\text{elem}[f_{ij}]$. érték = Adat akkor

listatörrend := listatörrend (f_{ij})

különben

$pe := f_{ij}$; $p := \text{elem}[f_{ij}]$. töv;

amíg $(p \leftrightarrow végjel)$ és $(\text{elem}[p]$. érték < Adat) ismét

$pe := p$; $p := \text{elem}[p]$. töv;

a vége;

ha $(p \leftrightarrow végjel)$ és $(\text{elem}[p]$. érték = Adat) akkor

listatörrend := listatörrend (pe)

különben

listatörrend := hamis

k vége ;

k vége

különben

listatörrend := hamis;

k vége ;

vége ;

III. Strázzás lista:

Strázzás lista inicializálása:

függvény slistarezd (fej, vég; mutatótípus); logikai;

ha elfoglal (fej) akkor

vég := fej;

slistarezd := igaz

különben

slistarezd := hamis;

ha vége;

vége;

Strázzás lista feldolgozása:

eljárás slistafeldolg (fej, vég; mutatótípus);

változó p; mutatótípus;

p := fej;

amíg p <> vég ismételt

elem [p]. érték feldolgozása;

p := elem [p]. köv;

a vége;

vége;

Keresés struktúra listában:

függvény lista keres (fej, vég: mutató típus; Adat: érték típus,
hely: mutató típus): logikai;

változó p: mutató típus;

elem [vég]: érték := Adat;

p := fej;

amíg elem [p]. érték <> Adat ismételt

p := elem [p]. köv;

a vége;

hely := p;

lista keres := (p <= vég);

vége;

Struktúra lista bővítése:

függvény lista felszélé (csellé, vég: mutató típus; Adat: érték típus): logikai;

változó p: mutató típus;

ha elfogad (p) akkor

elem [p]: = elem [csellé];

elem [csellé]. érték := Adat;

elem [csellé]. köv := p;

ha vég = csellé akkor

vég := p;

h. vége;

lista felszélé := igaz

különben

lista felszélé := hamis;

h. vége;

vége;

Elem törlése strázsás listából:

függvény $\text{listatöröl}(est, vég; \text{mutatótípus}): \text{logikai};$

változó $p: \text{mutatótípus};$

ha $est \wedge \neg vég$ akkor

$p := \text{elem}[est \div \text{lév}];$

$\text{elem}[est \div 1] := \text{elem}[p \div 1];$

ha $vég = p$ akkor

$vég := est;$

u vége;

fel szabadít (p);

listatöröl := igaz

különben

listatöröl := hamis;

u vége;

vége;

FA:

1. Bináris fa:

deklarálás:

konstans maxelem = maximális elemzés,

konstans végjel = \emptyset ;

típus mutatótípus: egész;

típus értéktípus: tárolandó-érték típusa;

típus binárisfaelemtípus: rekord (

érték: értéktípus;

bal, jobb: mutatótípus;

)

változó elem: tömb [1..maxelem] binárisfaelemtípus;

inicializálás:

eljárás binárisfaelem (gyökér: mutatótípus);

gyökér := végjel;

vége;

Bináris fa preorder bejárása:

eljárás binárisfaelem (p: mutatótípus);

ha p <> végjel akkor

elem [p].érték feldolgozása

binárisfaelem (elem [p].bal);

binárisfaelem (elem [p].jobb);

h vége;

vége;

Binaris fa inoder bejarsa:

ejars linfaivoder (p : mutatorpus);

ka $p \leftrightarrow$ vejel arbor

linfaivoder (elem f_{pT} . bal);

elem f_{pT} . etel feldolgozasa

linfaivoder (elem f_{pT} . jobb);

li veje;

veje;

Binaris fa postorder bejarsa:

ejars linfaipostorder (p : mutatorpus);

ka $p \leftrightarrow$ vejel arbor

linfaipostorder (elem f_{pT} . bal);

linfaipostorder (elem f_{pT} . jobb);

elem f_{pT} . etel feldolgozasa;

li veje;

veje;