Bináris fa nem rekurzív preorder bejárása:

```
eljárás bináris_preorder (p: mutatótípus);
  változó verem: veremtípus;
  változó p, f: mutatótípus;
  változó veremet: logikai;
  veremtörzsd (verem);
  veremet := verembe (verem, p) és verembe (verem, 1);
  amíg veremet és nem veremüres (verem) ismétel
    veremet := veremből (f) és veremből (p);
    ha veremet és (p <> végjel) akkor
      elágazás
        amikor f = 1:
          elem[p].érték feldolgozása
          veremet := verembe (verem, p) és verembe (verem, 2)
          veremet := verembe (verem, elem[p].bal) és
                     verembe (verem, 1);
        amikor f = 2:
          veremet := verembe (verem, elem[p].jobb) és
                     verembe (verem, 1);
      e vége;
    h vége;
  a vége;
vége;
```

Bináris fa nem rekurzív inorder bejárása:

```
eljárás binfanemrekin (p : mutatótípus);
  változó  verem : veremtípus;
  változó  p, f : mutatótípus;
  változó  veremok : egészek;
  veremreset (verem);
  veremok := verembe (verem, p) és verembe (verem, 1);
  amíg veremok és nem veremüres (verem) ismétel
    veremok := veremből (f) és veremből (p);
    ha veremok és (p <> végjel) akkor
      elágazás
        amikor f = 1:
          veremok := verembe (verem, p) és verembe (verem, 2);
          veremok := verembe (verem, elem fpJ.bal) és
                     verembe (verem, 1);
        amikor f = 2:
          elem fpJ.értéke feldolgozása
          veremok := verembe (verem, elem fpJ.jobb) és
                     verembe (verem, 1);
      e vége;
    h vége;
  a vége
vége;
```

Bináris fa nem rekurzív postorder bejárása:

```
eljárás binfanemrekpost (p: mutatótípus);
  változó  verem : veremtípus;
  változó  p, f : mutatótípus;
  változó  veremből : logikai;
  veremtörlés (verem);
  veremből := veremből (verem, p) és veremből (verem, 1)
  amíg veremből és nem veremüres (verem) ismétel
    veremből := veremből(f) és veremből (p);
    ha veremből és (p <> végjel) akkor
      elágazás
        amikor f = 1 :
          veremből := veremből (verem, p) és veremből (verem, 2);
          veremből := veremből (verem, elem[p].bal) és
                      veremből (verem, 1);
        amikor f = 2:
          veremből := veremből (verem, p) és veremből (verem, 3);
          veremből := veremből (verem, elem[p].jobb) és
                      veremből (verem, 1);
        amikor f = 3:
          elem[p].érték feldolgozása;
      e vége;
    ha vége;
  a vége;
vége;
```

II. Keresőfa:

Keresés a keresőfában:

```
függvény keresőfakeres (gyökér : mutatótípus; Adat : értéktípus;
                        hely : mutatótípus) : logikai;
változó  p : mutatótípus;
p := gyökér;
amíg (p <> végjel) és (elem[p].érték <> Adat) ismétel
  ha Adat < elem[p].érték akkor
    p := elem[p].bal
  különben
    p := elem[p].jobb;
  ha vége;
a vége;
hely := p;
keresőfakeres := p <> végjel;
vége;
```

Beszúrás keresőfába:

```
függvény keresőfabeszúr (gyökér : mutatótípus; Adat : értéktípus): logikai;
  változó  p, szülő, új : mutatótípus;
  p := gyökér;
  amíg (p <> végjel) és (elem[p].érték <> Adat) ismétel
    szülő := p;
    ha Adat < elem[p].érték akkor
      p := elem[p].bal
    különben
      p := elem[p].jobb;
    ha vége;
  a vége;
  ha p = végjel akkor
    ha lefoglal (új) akkor
      elem[új].érték := Adat;
      elem[új].bal := végjel; elem[új].jobb := végjel;
      ha gyökér <> végjel akkor
        ha Adat < elem[szülő].érték akkor
          elem[szülő].bal := új
        különben
          elem[szülő].jobb := új;
        ha vége
      különben
        gyökér := új;
      ha vége;
      keresőfabeszúr := igaz
```

```
különben
    keresőfabeszúr := hamis ;
ha vége
különben
    keresőfabeszúr := hamis;
ha vége;
vége;
```

Törlés keresőfából:

```
eljárás töröl ø ( gyökér, szülő, p : mutatótípus);
ha p = gyökér akkor
    gyökér := végjel
különben
    ha p = elem [szülő].bal akkor
        elem [szülő].bal := végjel
    különben
        elem [szülő].jobb := végjel;
    ha vége;
ha vége;
vége;
```