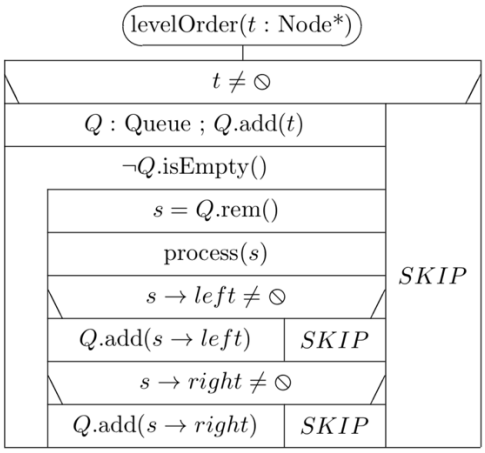
8. Gyakorlat (2019. 04. 02.)

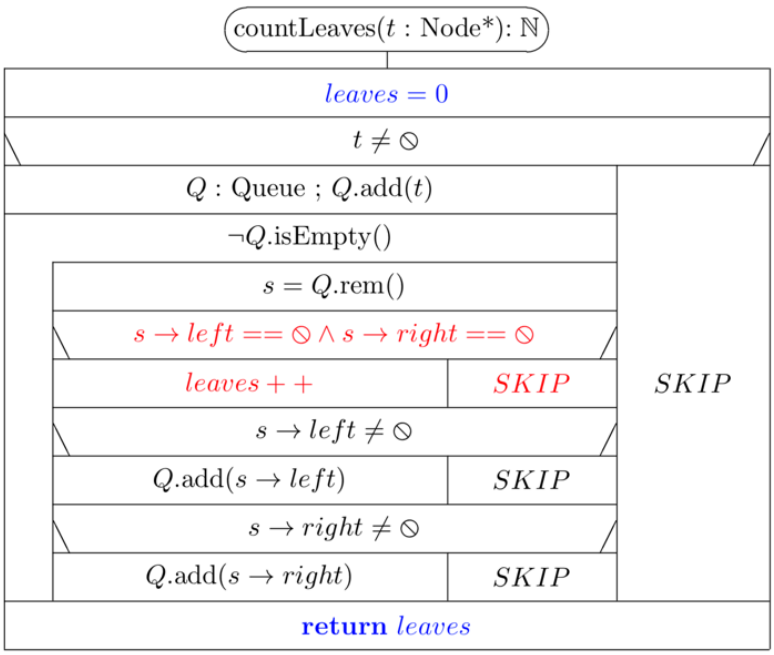
Szintfolytonos bejárás

A bináris fa csúcsait a mélységük szerinti sorrendben látogatjuk meg, egy szinten belül balról jobbra. Ehhez egy sort használunk, ez egy nagyon jó példa a Queue alkalmazására.

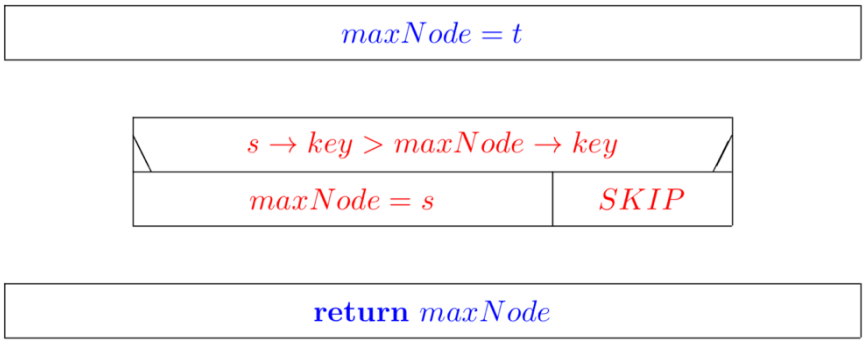


Műveletigénye Θ(n), ahol n a csúcsok száma, hiszen minden csúcs egyszer kerül bele a sorba, és minden iterációban kiveszünk egyet, amivel konstans műveletet végzünk.

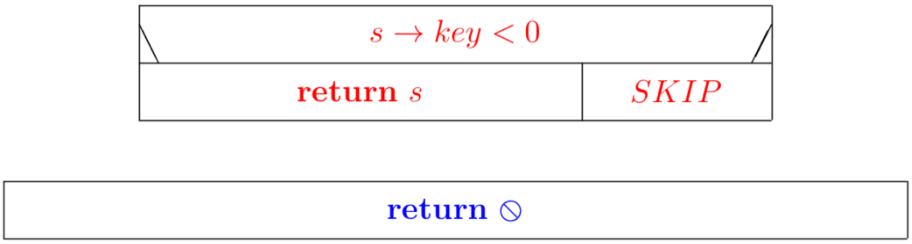
Feladat: számoljuk meg egy bináris fa leveleit szintfolytonos bejárással! Ez egy gyakorló feladat,a mit megodlottunk már rekurzívan is. Párhuzam vonható a megszámolás programozási tétellel.



Feladat: Maximális kulcsú elem keresése. Itt lényegében egy maximum kiválasztást ágyazunk a bejárásba.



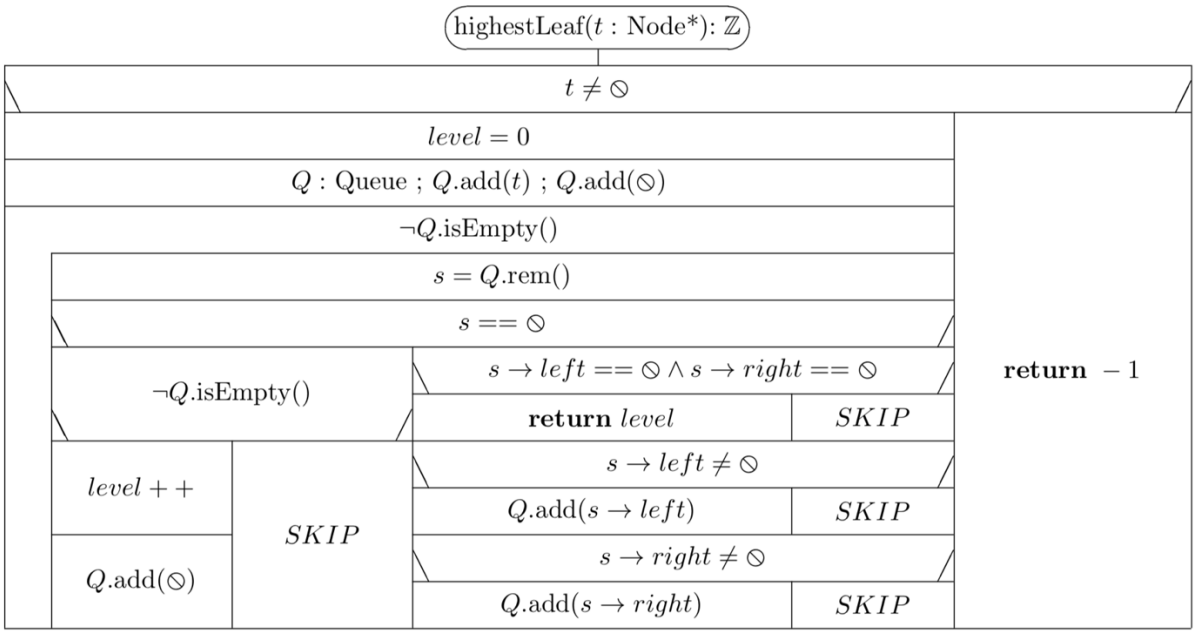
Feladat: negatív kulcsú elem keresése. Itt párhuzam vonható a keresés programozási tétellel. Ahhoz hasonlóan megoldható egy *while* ciklussal is, de talán szebb és könnyebben érthető, ha a ciklusból return-el lépünk ki.



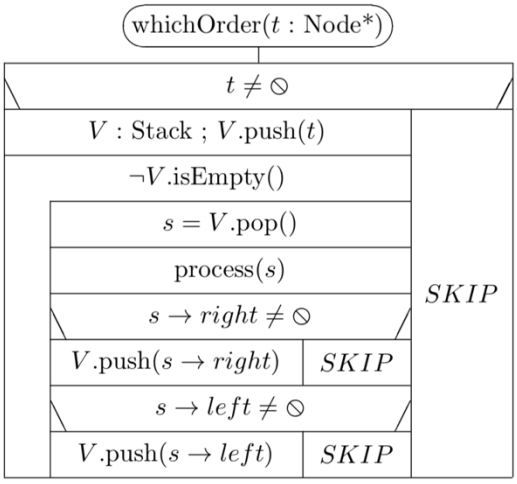
Feladat: hányadik szinten van a legfelső levél a fában? (Itt kifejezetten előnyös a szintfolytonos bejárás.)

Megállíthatjuk a bejárást, amint egy levelet találtunk. Viszont: hogyan számoljuk azt, hogy hányadik szinten vagyunk? Többféle megoldás is létezik:

- „Párokat" teszünk a sorba, minden csúcshoz berakjuk mellé a szintjét.
- Végjelekkel: minden szint végén egy végjelet (NULL) teszünk a sorba. Ezt úgy oldjuk meg, hogy kezdetben a gyökérelem után teszünk egy végjelet, majd amikor végjelet veszünk ki, megnöveljük a szintet, és beteszünk egy új végjelet a sorba.
- Szintek elemszámainak számolásával: a végjelek helyett számoljuk az elemeket. Mindig nyilvántartjuk azt, hogy
  - hány elem van ezen a szinten
  - a szinten belül hányadik elemnél tartunk
  - számláljuk a következő szint elemszámát



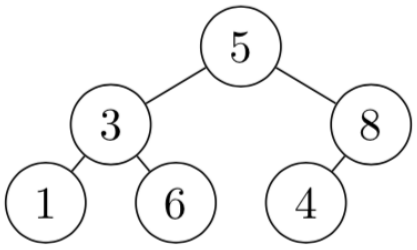
Módosítsuk úgy a bejárást, hogy sort helyett vermet használunk, és a gyerekek sorrendjét felcseréljük, előbb a jobb-, aztán a bal gyereket tesszük a verembe (ha van). Ekkor az egyik rekurzív bejáró algoritmus iteratív változatát kapjuk (Preorder).



Bináris keresőfa

A keresőfa tulajdonság:

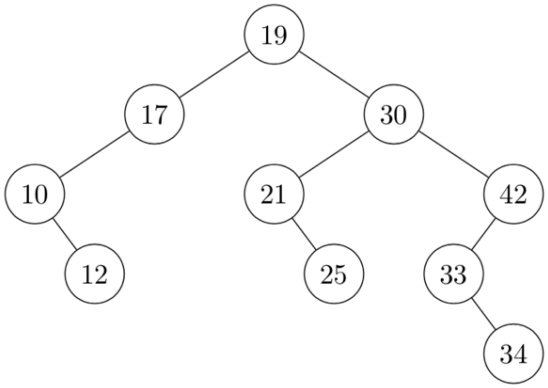
- Egy hibás definíció: minden csúcs balgyerekének kulcsa kisebb, jobbgyerekének kulcsa nagyobb. ROSSZ! A bal részfában egy szinttel mélyebben lehetne nagyobb kulcsú elem. Pl:



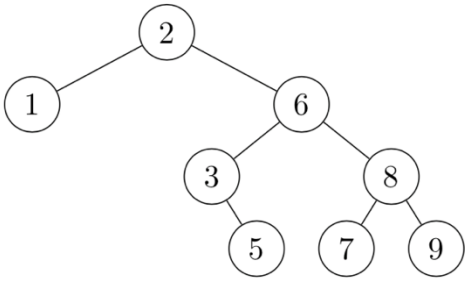
- Helyes definíció: egy csúcs kulcsánál a bal részfájában minden csúcs kulcsa kisebb, a jobb részfájában minden csúcs kulcsa nagyobb. Nem engedünk egyenlőséget, tehát a bináris keresőfában csupa különböző elemek vannak. **Rendezőfa** elnevezést használjuk, ha megengedjük az egyenlőséget.

Bináris keresőfa sorozatos beszúrásokkal:

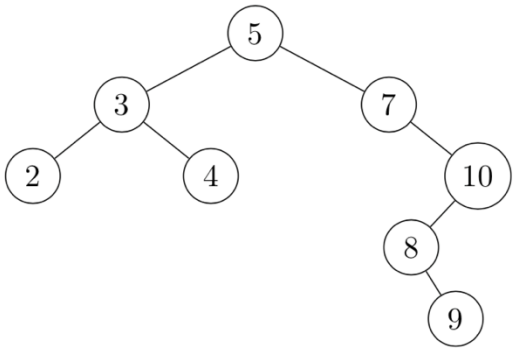
$$19, 17, 30, 10, 21, 12, 42, 25, 33, 34$$



Feladat: Egy bináris keresőfa preorder bejárásában a kulcsok sorrendje: 2, 1, 6, 3, 5, 8, 7, 9. Rajzoljuk fel a keresőfát. Megoldás: az első elem a gyökér, ezt követi a balgyereke, míg a jobbgyereke a sorrendben az első nála nagyobb elem. Ezt kell rekurzívan alkalmazni a részfákra is.



Feladat: Egy bináris keresőfa postorder bejárásában a kulcsok sorrendje: 2, 4, 3, 9, 8, 10, 7, 5. Rajzoljuk fel a keresőfát. A z utolsó elem a gyökér, ezt előzi meg a jobbgyerek, míg a balgyereke a sorrendben hátulról haladva az első nála kisebb elem. Ezt kell rekurzívan alkalmazni a részfákra is.

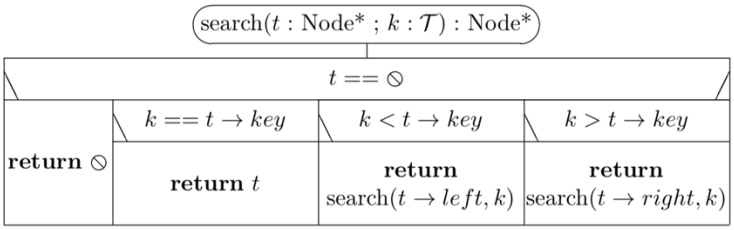


A bináris keresőfa alapműveletei

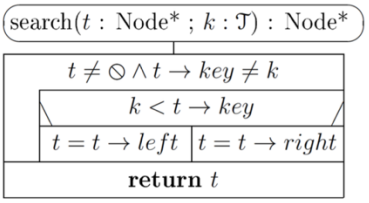
- *search(t, k)*: a fában megkeresi a k kulcsú elemet, NULL-t ad vissza, ha nincs benne.
- *insert(t, k)*: beszúrja a k kulcsú elemet, ha még nincs a fában.
- *min(t)*: minimális kulcsú csúcs címét adja vissza.
- *remMin(t, minp)*: a fából kiveszi a minimális kulcsú csúcsot, és a címét a minp pointerben adja vissza.
- *del(t, k)*: törli a k csúcs elemet, amennyiben az megtalálható a fában.

Mindegyik művelet O(h), ahol *h* a fa magassága.

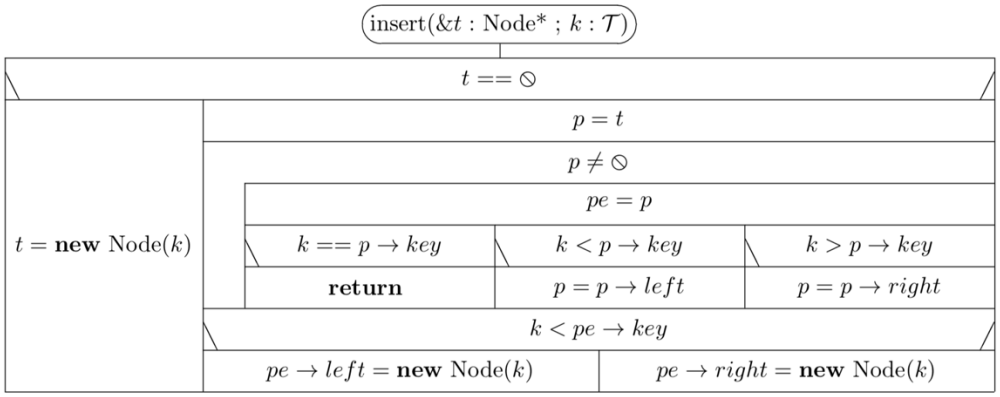
1. A keresés rekurzívan.



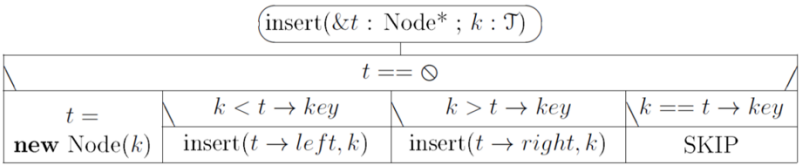
(A jegyzetbeli iteratív verzió:)



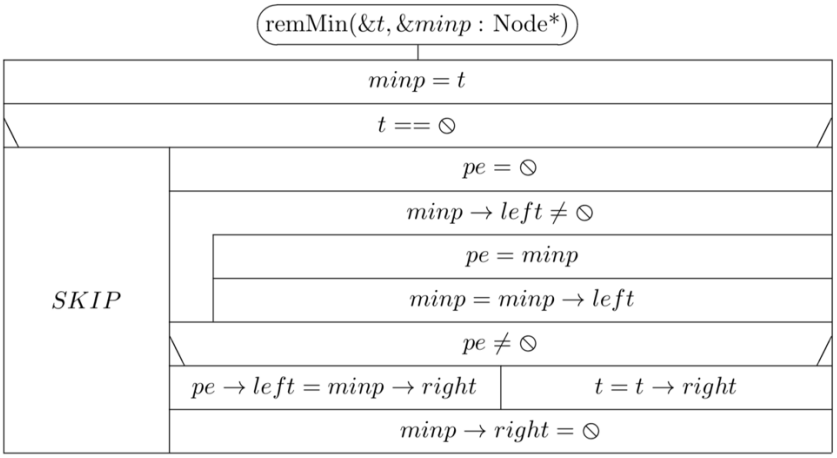
2. A beszúrás iteratívan.



(A jegyzetbeli rekurzív verzió:)



Minimum törlés iteratívan.



Házi feladatok:

1. A szintfolytonos bejárást alkalmazva határozzuk meg egy bináris fában azon levelek számát, amelyek mélysége legalább *k*. (Hint: gyakorlaton néztünk példát szintszámlálós algoritmusra, szóval csak annak a struktogramját kell kiegészíteni.)

2. Oldjuk meg a bináris keresőfa ellenőrzést úgy, hogy csak pozitív kulcsok vannak a fában, de semmi mást nem tehetünk fel a kulcsokról (nincs felső korlátjuk). Az inorder bejárás szerint a kulcsoknak szigorúan monoton növekedő sorrendben kell lenniük. A *k* cím szerinti paraméterben mindig az előző kulcsérték van.