



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

Algoritmusok és Alkalmazásaik Tanszék

Keresőfák és nevezetes algoritmusaikat szemléltető program

Témavezető:

Veszprémi Anna

Mestertanár

Szerző:

Ujj László

Programtervező Informatikus

Budapest, 2011

Tartalomjegyzék

1.	fejezet Bevezetés	2
2.	fejezet Felhasználói dokumentáció.....	3
2.1	A feladat rövid ismertetése.....	3
2.2	Célközönség	4
2.3	Rendszerkövetelmények	4
2.4	Első üzembe helyezés	4
2.5	Általános felhasználói tájékoztató	5
2.6	A rendszer funkcióinak ismertetése.....	6
2.6.1	Bináris keresőfa.....	6
2.6.2	AVL-fa	9
2.6.3	2-3 fa	12
2.6.4	A Fájl menü.....	15
2.6.5	A Szerkesztés menü.....	16
2.6.6	A Műveletek eszköztár	17
2.6.7	Az Animáció eszköztár.....	19
2.6.8	Helyi menük	21
2.6.9	Animáció rajzolási területe	22
2.6.10	Algoritmus megjegyzés területe	22

1. fejezet

Bevezetés

Az „Algoritmusok és adatszerkezetek” tantárgy első félévében szerepelnek a keresőfák, és nevezetes algoritmusai. Szakdolgozatomban egy olyan programot készítettem, amellyel a tananyagban szereplő keresőfa adatszerkezetek felépítését és működését lehet tanulmányozni. A program ennek az anyagrésznek az elsajátítását segíti, grafikus felületen ábrázolja a kiválasztott adatszerkezeteket, és szemlélteti rajtuk a tanult algoritmusok működését.

A keresőfák olyan adatszerkezetek, melyekkel hatékonyan lehet dinamikusán változó halmazokat kezelni. Tipikus műveleteik: *keresés*, *beszúrás*, *törlés*, *maximum*, *minimum*, *előző* és *következő*. A műveletek hatékonysága a fa magasságával arányos.

Három fajta keresőfát szemléltet a program:

- bináris keresőfa
- AVL fa
- 2-3 fa

A bináris keresőfában minden csúcsnak két gyereke lehet. A műveletek akkor végezhetők el a leghatékonyabban, ha alakja közelít a teljes bináris fához. Ez a dinamikusán változó kulcshalmaz miatt nem biztosítható. Erre ad egy megoldást az AVL fa, mely a műveletek után forgatásokkal igyekszik a fa alakját kiegyensúlyozott helyzetben tartani. A 2-3 fák azért fontosak, mert rajtuk lehet tanulmányozni, és megérteni, hogy milyen elven működnek az adatbázisrendszerek által használt B-fák.

A felhasználó egy könnyen kezelhető grafikus felület segítségével irányíthatja a műveleteket, és tanulmányozhatja a szemléltetett adatszerkezetek működését.

2. fejezet

Felhasználói dokumentáció

2.1 A feladat rövid ismertetése

Feladatom az volt, hogy könnyen kezelhető, látványos, jól szemléltető programot készítsék a keresőfák szerkezetének, működésének tanulmányozásához.

A bináris keresőfa egy olyan keresőfa, melyben minden csúcspontnak legfeljebb 2 gyermeke van és tetszőleges x csúcsára igaz, hogy x bal oldali részfájában található valamennyi elem kulcsa kisebb kulcs(x)-nél, a jobb oldali részfa elemeinek kulcsai pedig nagyobbak. Ez alapján egy bináris keresőfa minden részfája is bináris keresőfa. A fa inorder bejárása a csúcsokban tárolt kulcsok szerint növekvőleg látogatja meg az elemeket. Általában a csúcspontok által reprezentált információ rekord, nem pedig egyetlen adat. Azonban, összehasonlításkor, a csúcspontok kulcsai vannak figyelembe véve, mintsem a hozzájuk kapcsolódó rekordok bármely része.

A bináris keresőfán a keresésen kívül értelmezzük még a beszúrás, törlés műveleteket. Adott csúcspontnak kereshetjük az előzőjét és következőjét is. A fának megkereshetjük a legkisebb kulcs-értéket tároló csomópontját, és a legnagyobbat is. A fát bejárhatjuk adott sorrendben is. Egy fa bejárásán egy olyan folyamatot értünk, amikor a fa összes csúcsát meghatározott sorrendben látogatjuk meg (megvizsgáljuk, vagy frissítjük), pontosan egyszer.

A műveletek hatékonysága függ a fa magasságától, az pedig attól, hogy mennyire kiegyensúlyozott a fa. Véletlenszerű beszúrások esetén is nagyon „megnyúlhat” a fa néhány ága, lerontva ezzel a keresés hatékonyságát. Egy n csúcsú teljes bináris fában a műveletek legrosszabb esetben $\Theta(\lg n)$ végrehajtási idejűek, ha viszont a fának egyetlen hosszú ága van, ez $\Theta(n)$ lesz. Ennek kiküszöbölésére ad megoldást az AVL fa.

Az AVL-fa alatt egy ön-kiegyensúlyozó bináris keresőfát értünk. Ez volt az első ilyen adatszerkezet, melyet feltaláltak. 1963-ban G. M. Adelson-Velsky és E. M. Landis publikálták. Egy AVL-fában bármely csúcspont két részfájának magassága közti különbség legfeljebb egy lehet. Minden csúcsnak van egy ún. egyensúly-faktora, amit megkapunk, ha kivonjuk a jobb részfa magasságát a bal részfából. Egy csúcs kiegyensúlyozott, ha ez az érték 1, 0 vagy -1. Minden más esetben a csúcs kiegyensúlyozatlan és forgatásokat igényel.

Az egyes műveletek megegyeznek egy kiegyensúlyozatlan bináris fán végrehajtottakkal, de kiegészülhetnek egy vagy több forgatással, amik a fa egyensúlyát hivatottak megtartani. A fa magassága n csúcs esetén legfeljebb $1,44 * \lg n$ lehet.

A 2-3 fák fontos jelentősége, hogy belőlük fejlődtek ki a B-fák. A 2-3 fákban minden belső csúcsnak 2 vagy 3 gyermeke van. A levelek egy szinten helyezkednek el. Az adatrekordok/kulcsok csak a levelekben vannak tárolva. A belső pontokban a keresést informáló kulcsok és mutatók vannak. A levelekben balról jobbra nőnek a kulcsok. 1 elemű fa esetén kivételesen a gyökérnek egy gyermeke van. A 2-3 fákon végrehajtott beszúrás és törlés után előfordulhat, hogy egy csomópontnak 3-nál több, vagy 2-nél kevesebb gyermeke lesz. Ekkor egy vagy több csúcsot is fel kell osztanunk, vagy össze kell vonnunk, hogy helyreállítsuk a fa tulajdonságait.

A fa magassága (h) és elemszáma (n) közötti összefüggés a következő: $2^h \leq n \leq 3^h$ és $h \leq \log_2 n$

2.2 Célközönség

A program elsősorban azoknak készült, akik épp az „Algoritmusok és adatszerkezetek” tantárgy első félévét hallgatják, de érdekes lehet mindazoknak, akik az anyagban szereplő keresőfák, és nevezetes algoritmusai után érdeklődnek, azokat szeretnék elsajátítani.

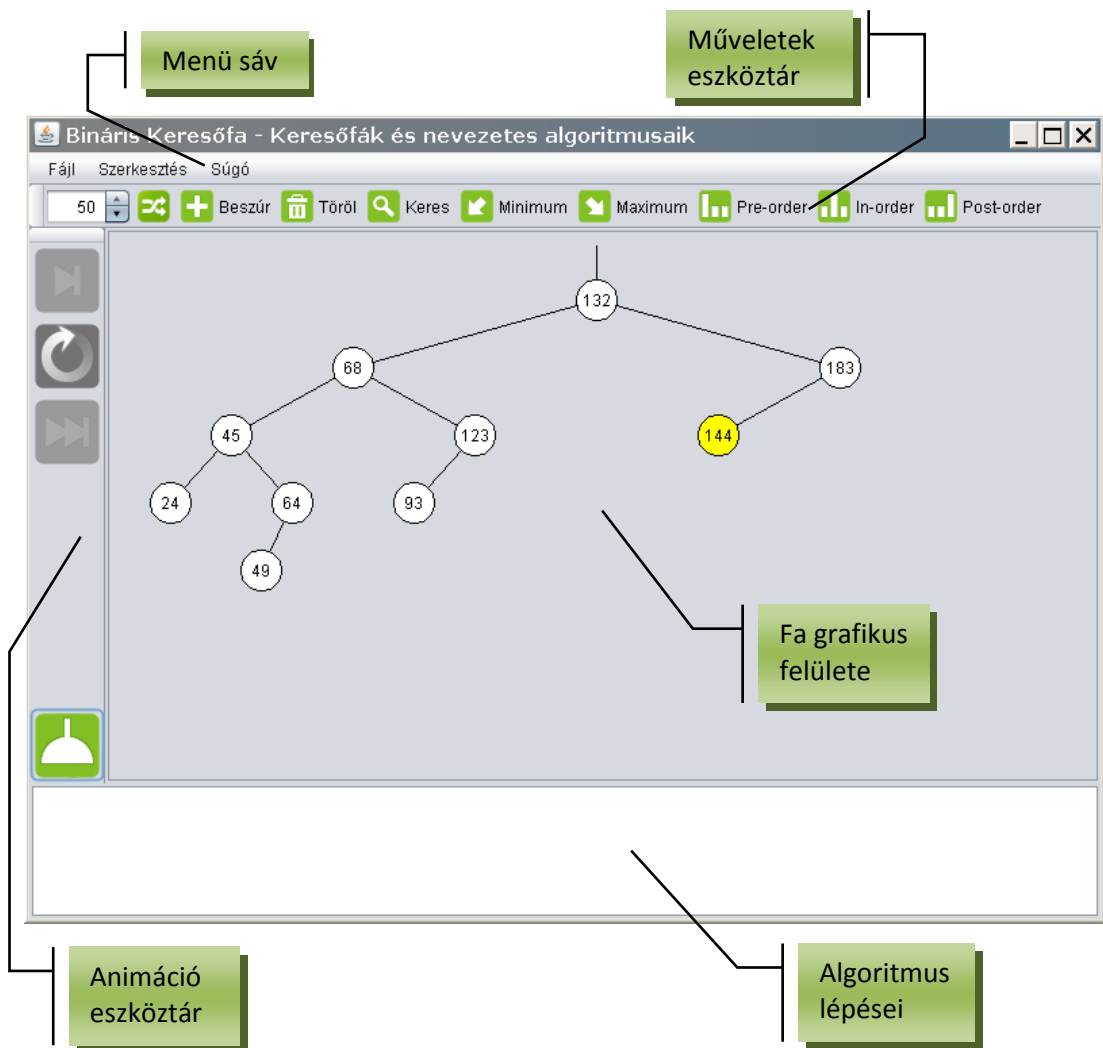
2.3 Rendszerkövetelmények

A szoftver java platformon fut. Használatához legalább 1.4-es verziójú Java™2 futtatókörnyezet szükséges.

2.4 Első üzembe helyezés

A szoftver nem igényel telepítést, közvetlenül CD-ről futtatható.

2.5 Általános felhasználói tájékoztató



A fájl menüben találhatóak egy új fa létrehozásához szükséges menüpontok. Szintén itt tudjuk elmenteni az aktuális fát, vagy betölteni egy régebben elmentettet. A szerkesztés menüben tudjuk feltölteni gyorsan a fát véletlen értékekkel, illetve itt tudjuk megnézni az aktuális fa zárójelezett alakját. A súgó menüben található a program névjegye.

Az menüsáv alatt található a műveletek eszköztár. Az eszköztár értékválasztó mezőjébe kell bevinni azt a kulcs-értéket, amivel dolgozni szeretnénk. Az eszköztár gombjai segítségével hajthatjuk végre a kívánt műveleteket a fán. Ha a beszúrást, törlést vagy keresést választjuk, a kívánt művelet az értékválasztó mezőben található értéket fogja kulcs-értékként használni. Az alkalmazás bal oldalán található az animáció eszköztár. Itt léptethetjük az aktuális animációt. Illetve állíthatjuk a fa csúcspontjainak elrendezését.

A fa grafikus felületén követhetjük nyomon az aktuális fa csúcspontjainak helyzetét, egymáshoz való viszonyát. A megfelelő csúcspontra jobb egérgombbal kattintva hozhatjuk elő a helyi menüt. Az alkalmazás alján olvashatjuk az aktuális algoritmus lépéseit.

Az egyes ikonok és menüpontok felett tartva a mutatót, egy rövid leírás jelenik meg az adott elem funkciójáról a mutató felett.

2.6 A rendszer funkcióinak ismertetése

2.6.1 Bináris keresőfa

2.6.1.1 Definíció

Legyen t egy bináris keresőfa, ekkor egy x csúcs bal oldali leszármazottait $bal(x)$, jobb oldali leszármazottait $jobb(x)$ -szel jelöljük.

A t bináris keresőfa bármely x csúcsára és $bal(x)$ bármely y csúcsára és $jobb(x)$ bármely z csúcsára igaz:

$$y \leq x \leq z, \text{ azaz } y < x < z \text{ (könnyítés)}$$

[1][6][7]

2.6.1.2 Műveletek

2.6.1.2.1 Keresés

A keresőfában keresi a k kulcsú elemet (csúcsot). A gyökértől indulva, mindig az adott csúcsban lévő értékkel összehasonlítjuk a keresett kulcsot. Ha megtaláltuk, visszatérünk a címével, ha nem, folytatjuk a keresést balra, ha a keresett kulcs kisebb, mint az aktuális, jobbra, ha nagyobb. Ha a megadott irányban nincs már leszármazott a keresés leáll, sikertelen jelzéssel visszatér. [1][5]

2.6.1.2.2 Maximum

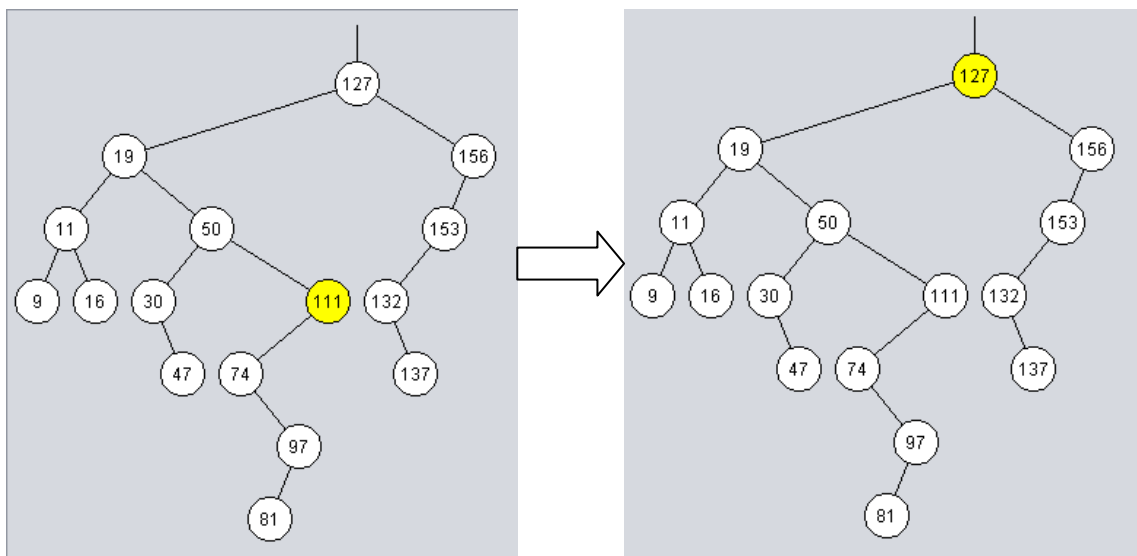
A maximális kulcsú elemet adja. A gyökértől indulva, amíg az aktuális csúcspontnak van jobb gyermeke, rálépünk. Ha nem tudunk tovább lépni, a keresés leáll, és jelzi melyik volt az utolsó csúcspont, amire rálépett. Üres fa esetén a keresés sikertelen. [1][5]

2.6.1.2.3 Minimum

A minimális kulcsú elemet adja. A gyökértől indulva, amíg az aktuális csúcspontnak van bal gyermeke, rálépünk. Ha nem tudunk tovább lépni, a keresés leáll, és jelzi melyik volt az utolsó csúcspont, amire rálépett. Üres fa esetén a keresés sikertelen. [1][5]

2.6.1.2.4 Következő

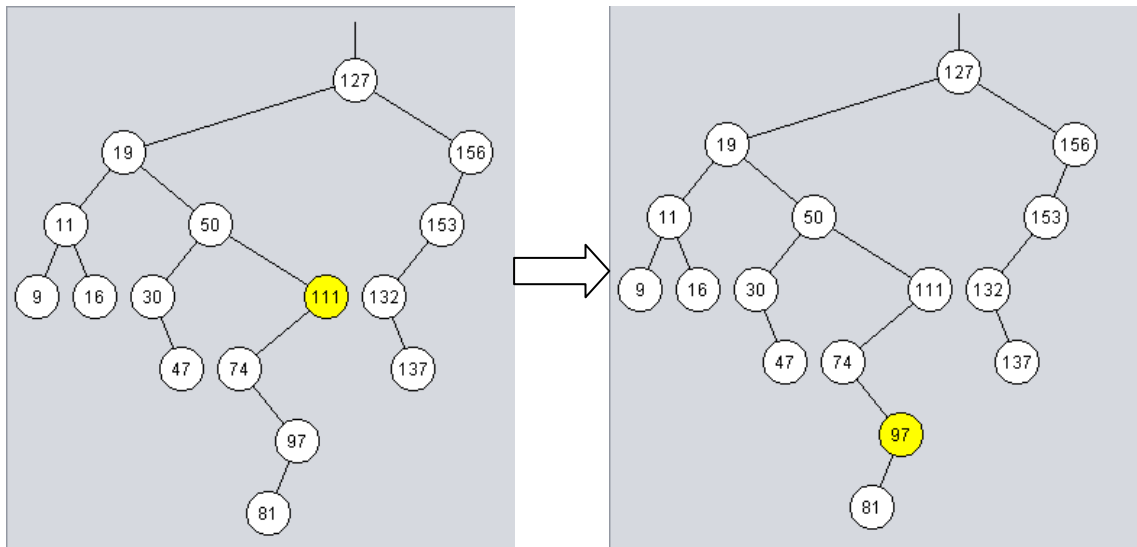
Adott csúcspont rákövetkezőjét adja, feltéve, ha van ilyen. Ha a kiválasztott csúcspontnak van jobb oldali részfája, akkor a keresés a jobb oldali részfa minimumát adja vissza. Ha a kiválasztott csúcspontnak nincs jobb oldali részfája, akkor addig lépünk a fában felfelé a szülőkön, amíg az adott szülőnek a kiválasztott csúcspont a jobb oldali részfájában van. A keresés megáll, ha olyan szülőt találunk, melynek a bal oldali részfájában van a kiválasztott csúcspont, és jelzi ezt a szülőt. A keresés sikertelen, ha már nincs hova lépni, és még mindig nem találtunk megfelelő szülőt. (Ez az az eset, amikor a maximális értékű kulcs rákövetkezőjét keressük.) [1][5]



1. ábra Adott elem következője bináris keresőfában.

2.6.1.2.5 Előző

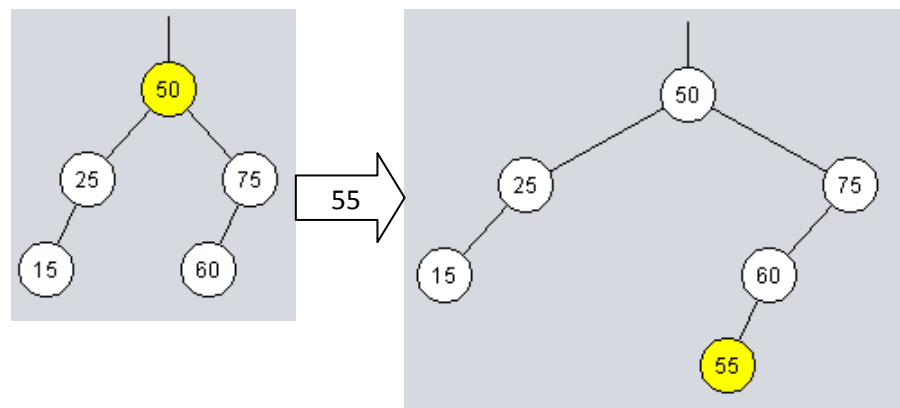
Adott csúcspont előzőjét adja, feltéve, ha van ilyen. Ha a kiválasztott csúcspontnak van bal oldali részfája, akkor a keresés a bal oldali részfa maximumát adja vissza. Ha a kiválasztott csúcspontnak nincs bal oldali részfája, akkor addig lépünk a fában felfelé a szülőkön, amíg az adott szülőnek a kiválasztott csúcspont a bal oldali részfájában van. A keresés megáll, ha olyan szülőt találunk, melynek a jobb oldali részfájában van a kiválasztott csúcspont, és jelzi ezt a szülőt. A keresés sikertelen, ha már nincs hova lépni, és még mindig nem találtunk megfelelő szülőt. (Ez az az eset, amikor a minimális értékű kulcs előzőjét keressük.) [1][5]



2. ábra Adott elem előzője bináris fában.

2.6.1.2.6 Beszúrás

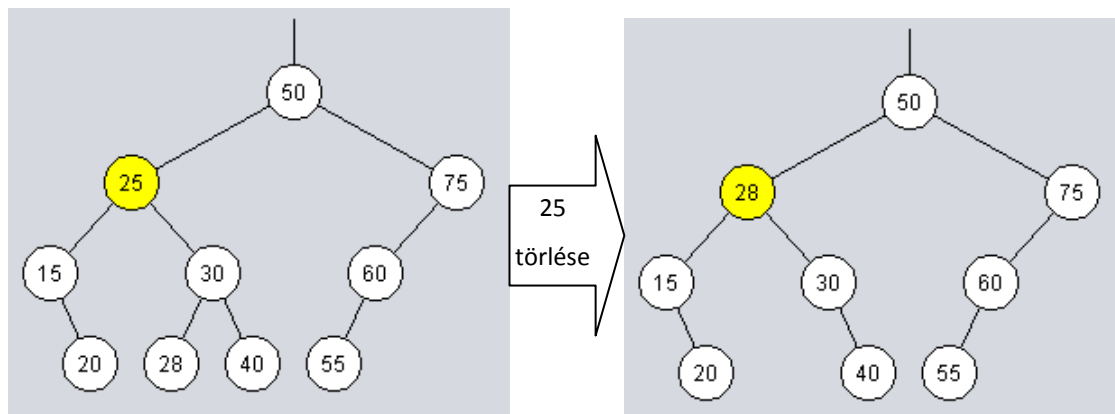
Beszúrja az adott értéket a fába, feltéve, hogy még nincs benne. A gyökértől indulva, mindig az adott csúcsban lévő értékkel összehasonlítjuk a beszúrandó kulcsot. Ha megtaláltuk, a kulcs már szerepel a fában, ezért nem kell újra beszúrni. Ha nem, folytatjuk a keresést balra, ha a beszúrandó kulcs kisebb, mint az aktuális, jobbra, ha nagyobb. Ha a megadott irányban nincs már leszármazott, akkor beszúrhatjuk a kulcsot az utoljára érintett csúcspont bal gyermekének, ha a beszúrandó kulcs kisebb, mint az aktuális, vagy jobb gyermekének, ha nagyobb. [1][5]



3. ábra Beszúrás bináris keresőfába.

2.6.1.2.7 Törlés

Adott kulcsot töröl a fából. Törléskor először megkeressük a törlendő kulcsot, ha nem szerepel a fában, akkor a művelet leáll, nem kell törölni. Ha szerepel, és nincs gyermeke, akkor ki kell törölni a fából. Ha van 1 gyermeke, akkor törlése után gyermekének a szülője a törölt elem szülője lesz. Ha 2 gyermeke van, akkor a rákövetkezőjét felhozzuk a helyére. A rákövetkező csúcs biztosan levél vagy egy-gyerekes csúcs. Ha gyereke van, csak jobb gyerekekkel rendelkezhet, hiszen a részfában ő a legkisebb. Ha levél, akkor egyszerűen töröljük a helyéről, majd a kitörölt elem helyére befűzzük. Ha egy-gyerekes, akkor a törlését az egy-gyerekes csúcs törlésénél megadottak szerint végrehajtjuk, majd befűzzük a törölt elem helyére. [1][5]



4. ábra Törlés bináris keresőfából.

2.6.2 AVL-fa

2.6.2.1 Definíció

Jelöljük $h(t)$ -vel a t bináris keresőfa magasságát.

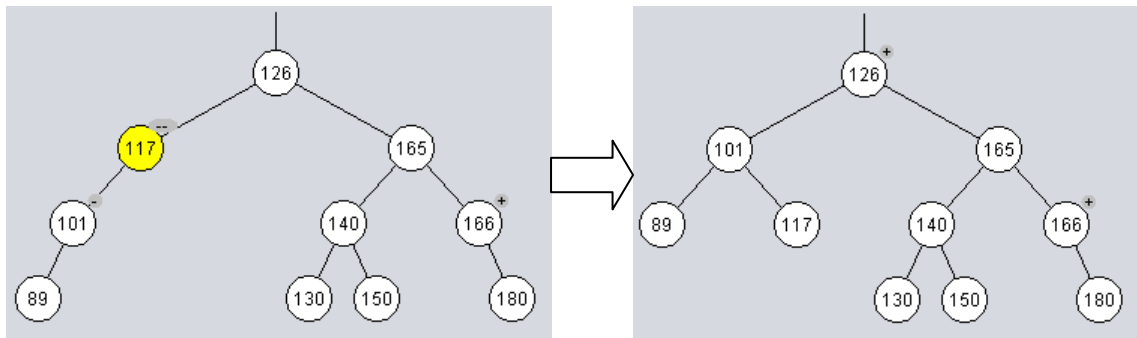
t bináris keresőfa kiegyensúlyozott (AVL-tulajdonságú) $\Leftrightarrow t$ minden x csúcsára igaz:

$$|h(bal(x)) - h(jobb(x))| \leq 1$$

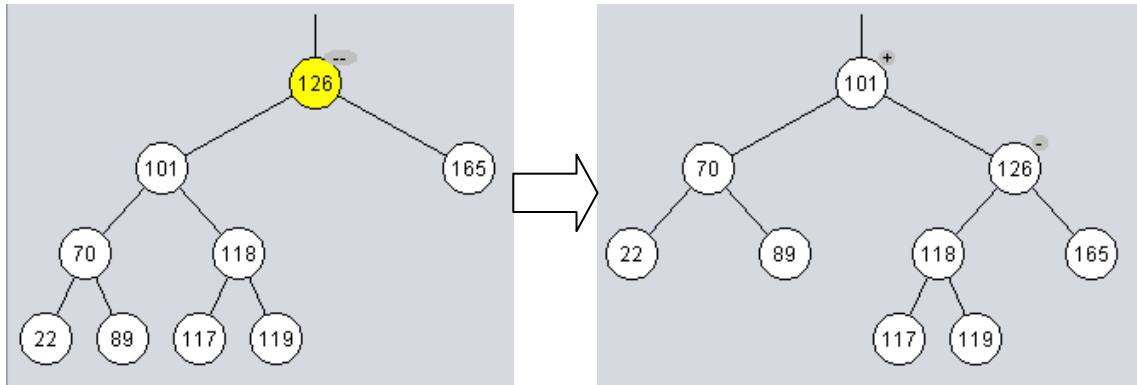
[2][6][7]

2.6.2.2 Műveletek

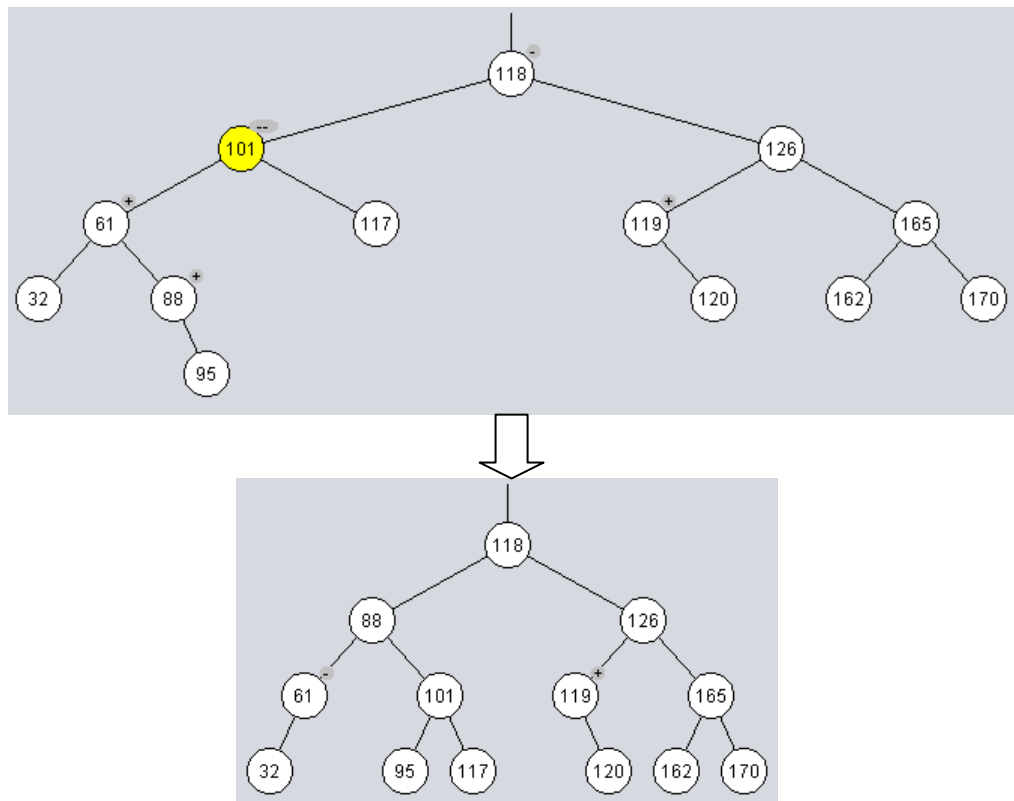
Műveletei azonosak a bináris keresőfa műveleteivel, így azokat nem sorolom fel újra. Viszont minden olyan művelet után, ami elronthatja a fa kiegyensúlyozottságát (beszúrás és törlés) ellenőrizzük, és ha kell, helyreállítjuk az AVL-tulajdonságot a megfelelő forgatással. [2]



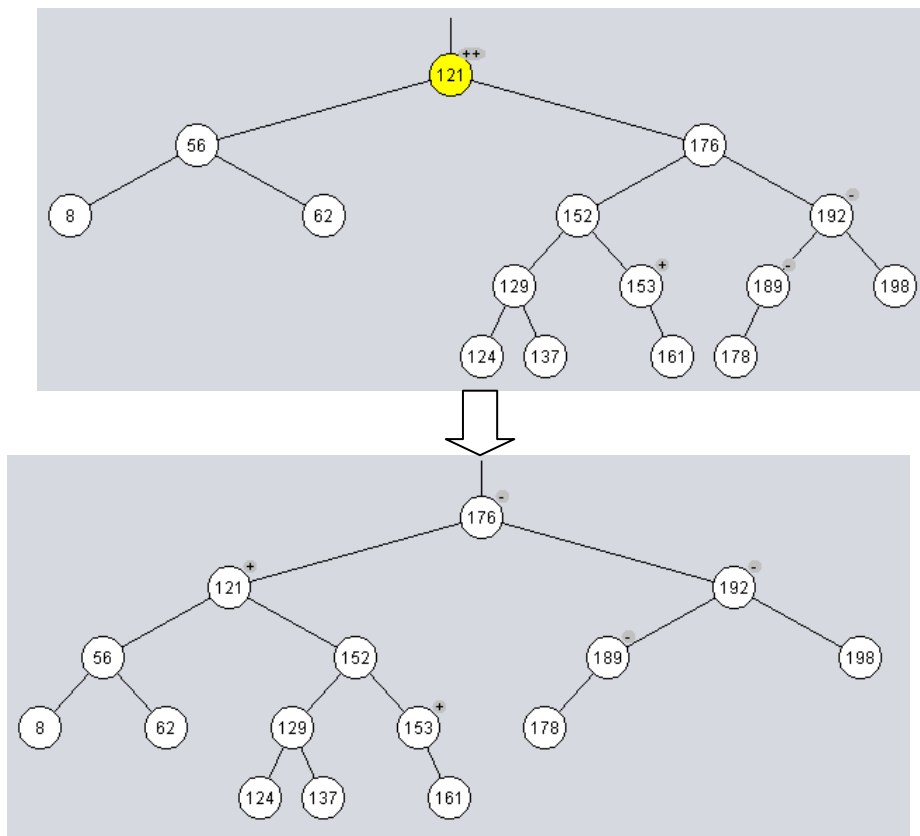
5. ábra (--,-) forgatás AVL-fában.



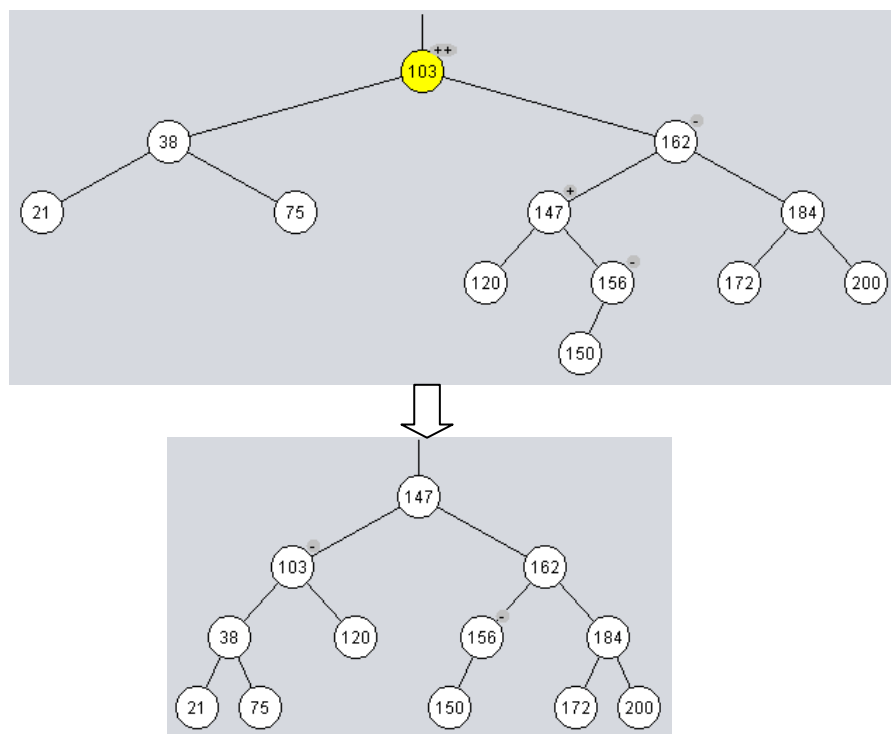
6. ábra (--,0) forgatás AVL-fában.



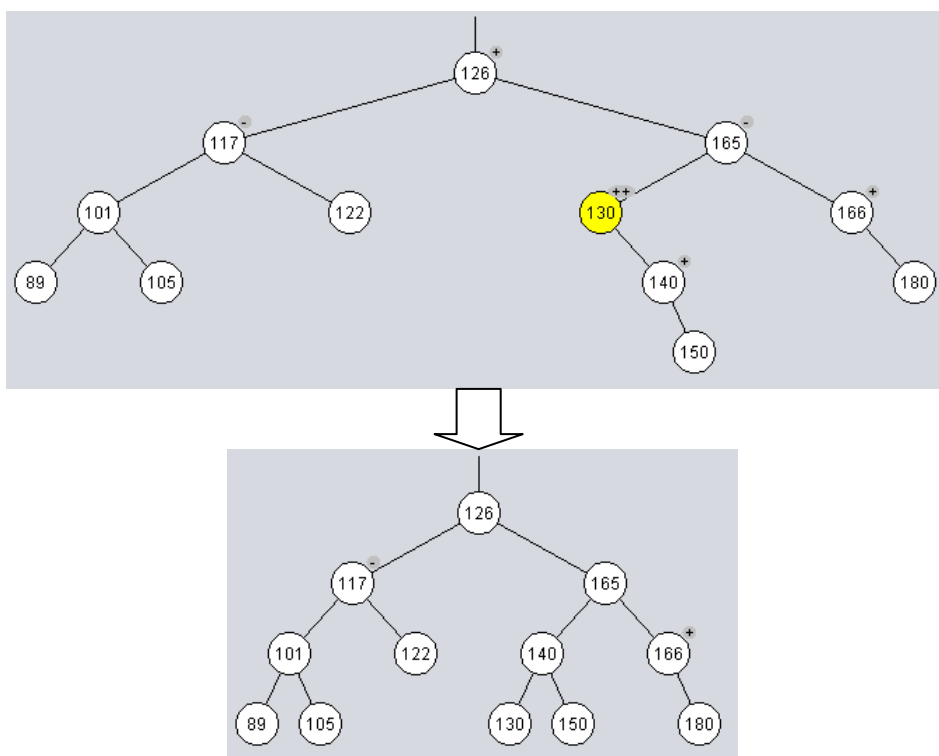
7. ábra (--,+) forgatás AVL-fában.



8. ábra $(++,0)$ fogatás AVL-fában.



9. ábra $(+,-)$ forgatás AVL-fában.



10. ábra (+++) forgatás AVL-fában.

2.6.3 2-3 fa

2.6.3.1 Definíció

Minden belső csúcsnak 2 vagy 3 gyermeke van. A levelek egy szinten helyezkednek el. Az adatrekordok/kulcsok csak a levelekben vannak tárolva. A belső pontokban a keresést informáló kulcsok és mutatók vannak. A levelekben balról jobbra nőnek a kulcsok. [3][6]

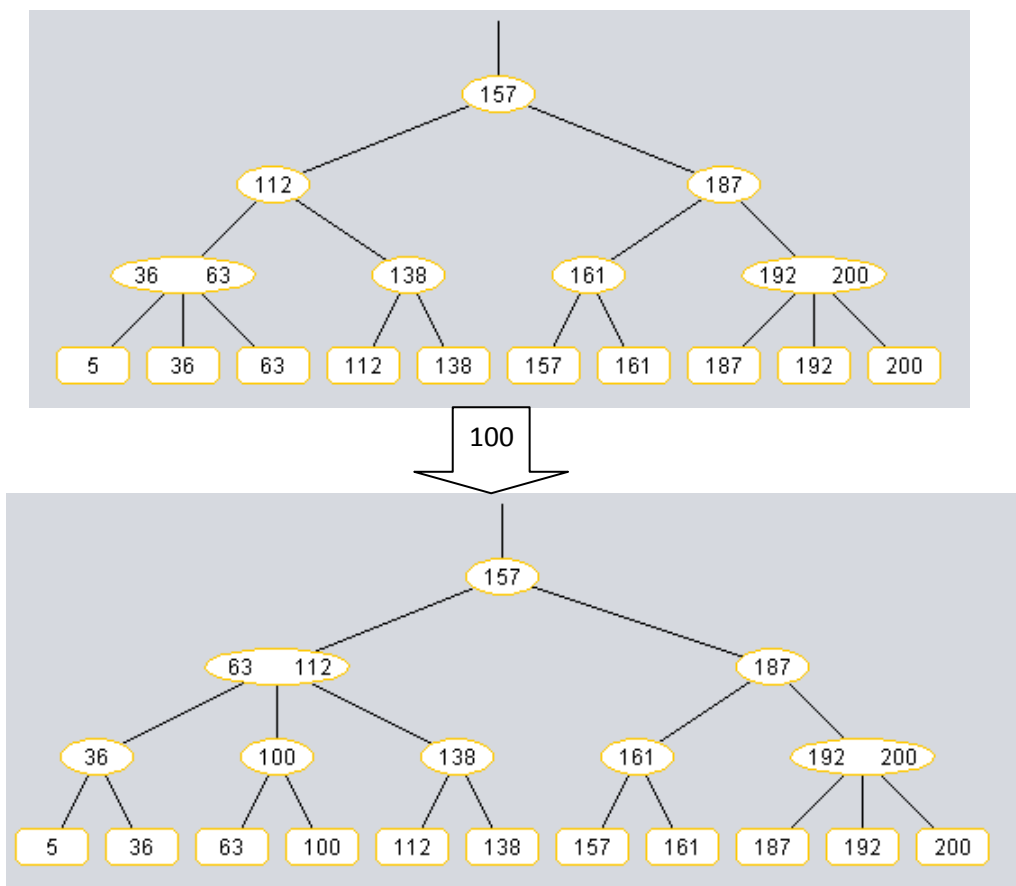
2.6.3.2 Műveletek

2.6.3.2.1 Keresés

A gyökértől indulva, mindig az adott csúcsban lévő első kulcs-értékekkel összehasonlítjuk a keresett kulcsot. Ha az adott csúcsnak csak 1 leszármazottja van, akkor arra lépünk. Ha két leszármazottja van, és keresett érték kisebb, mint az adott csúcsban lévő csúcsérték, akkor a bal oldali leszármazottra lépünk, ha nagyobb, akkor a jobb oldalra. Ha három leszármazottja van és a keresett érték kisebb a csúcsban lévő első kulcsnál, akkor a bal oldali leszármazottra, ha a két kulcs érték között van, akkor a középső leszármazottra, ha pedig mindkét kulcsnál nagyobb a keresett, akkor a jobb oldali leszármazottra lépünk. Addig haladunk, amíg levelet nem találunk. Ha a levél a keresett kulcsot tartalmazza, a keresés sikerrel járt, egyébként nem. [3][5]

2.6.3.2.2 Beszúrás

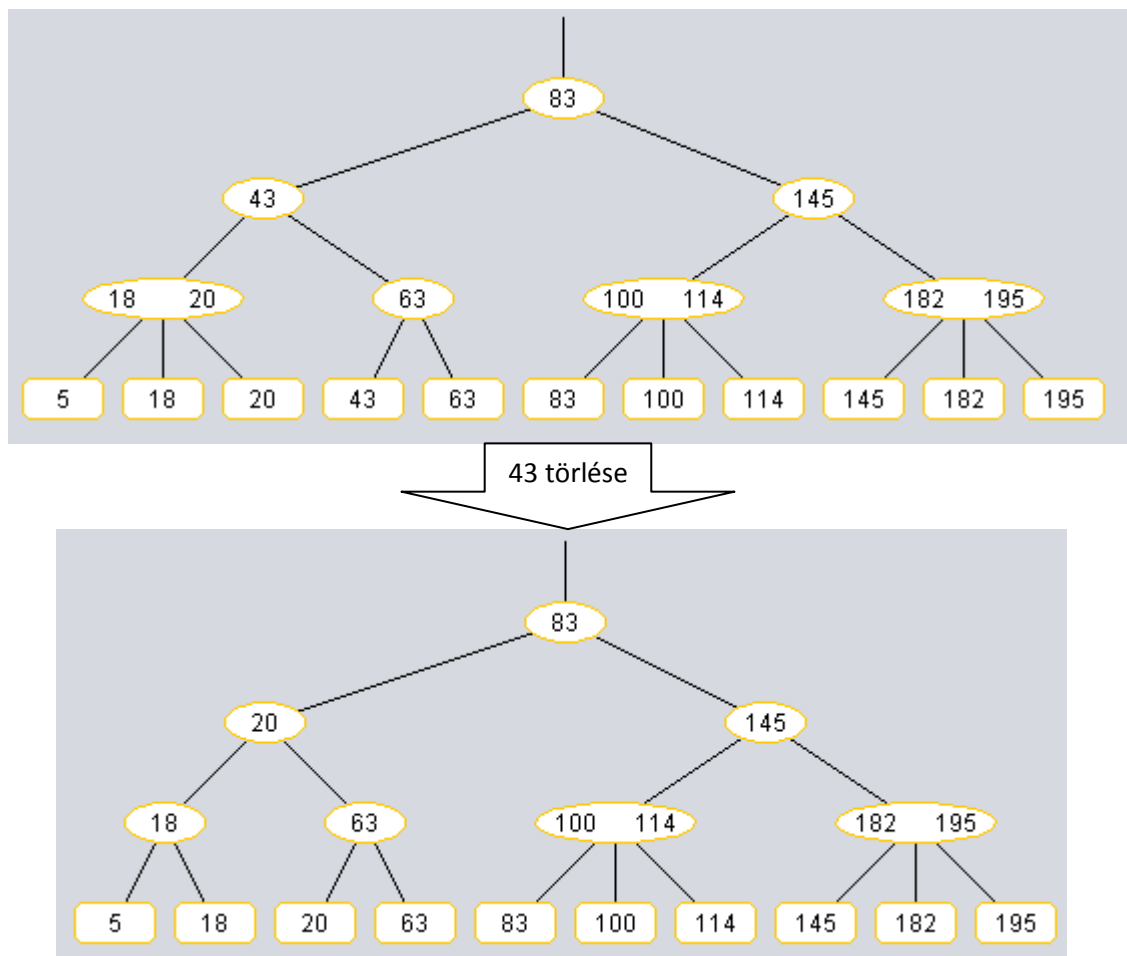
Beszúráskor megkeressük a beszúrandó kulcs-érték helyét a fában. Ha a kulcs már szerepel a fában, akkor megáll az algoritmus. Ha nem szerepel, akkor a keresés során legutoljára érintett nem levél csomópontba beszúrjuk a kulcs értéket. Ekkor előfordulhat, hogy ennek a csomópontnak már három leszármazottja van, így egy negyedik már nem fér bele. Ekkor szétvágjuk a csomópontot és két-két leszármazottját elosztjuk az új csomópontok között. Ha a szülőnek eleve 3 gyereke volt, akkor itt is csúcsvágásra van szükség, és így tovább felfelé. Ha valahol ezen az úton van egy 2 gyermekes belső csúcs, akkor ott megáll a beszúrás, mert annak lehet 3 gyereke. Ha ezen az úton minden belső pontnak 3 gyereke volt, akkor a csúcsvágás felgyűrűzik a gyökérig. Ekkor a kétfelé vágott gyökér fölé egy új gyökeret kell tenni. [3][5]



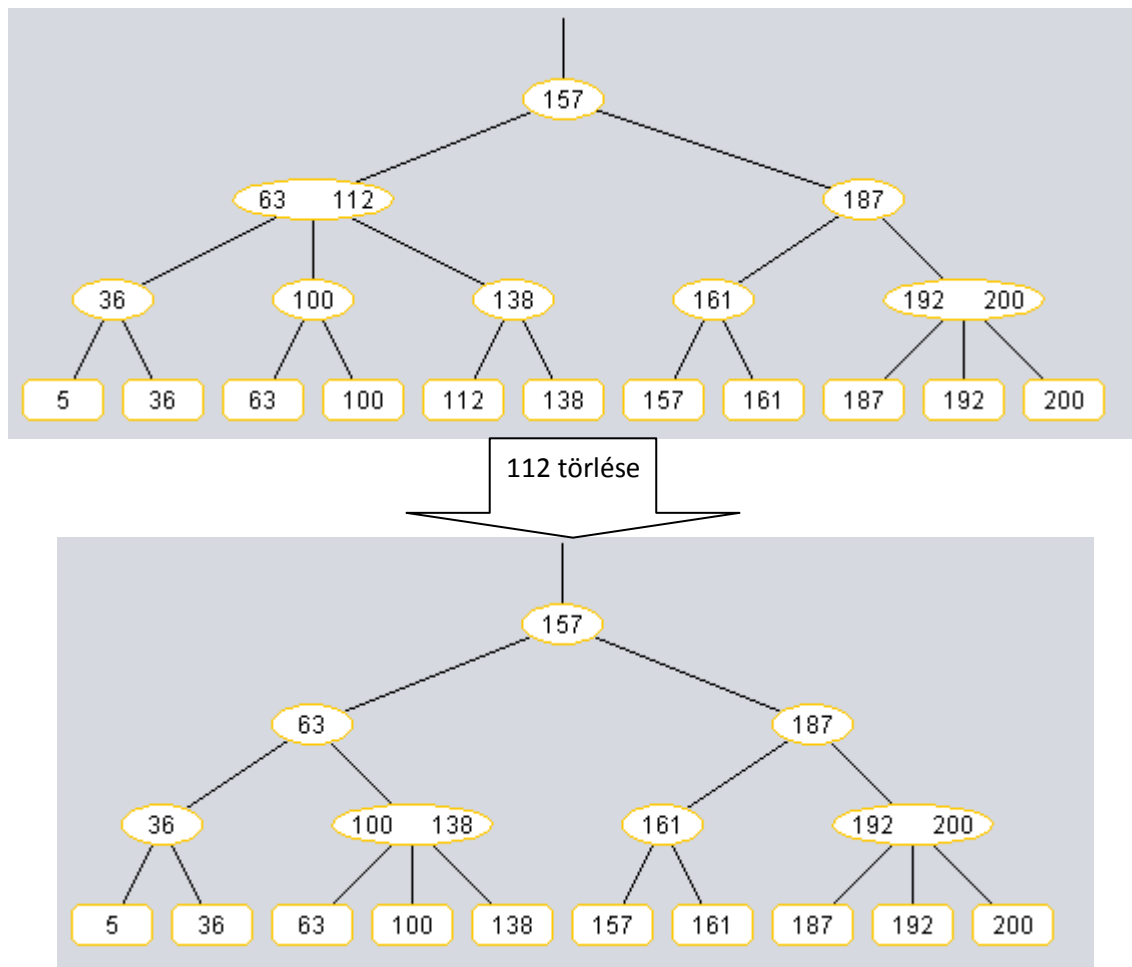
11. ábra Csúcsvágás 2-3 fában.

2.6.3.2.3 Törlés

Törlésnél megkeressük a törlendő kulcsot tartalmazó levelet a fában. Ha nem találjuk, akkor megáll a törlés. Ha a törlendő elem szülőjének 3 gyermeke van, akkor töröljük a szülőből a levelet. Ha a törlendő elemnek csak 1 testvére van, azaz a szülőnek 2 gyereke van és az ő szülőjének van 3 gyerekes testvére, akkor az 1 gyereket átad. Ha az ő szülőjének nincs 3 gyerekes testvére akkor a levél szülőjét összevonjuk egy testvérével. Szükség esetén a csúcsösszevonásokat folytatjuk felfelé. Ez felgyűrűzhet a gyökérig. Ha a gyökérnek ez után csak 1 gyermeke van, és az nem levél, akkor ez a gyermek lesz az új gyökér, és csökken a fa magassága. [3][5]



12. ábra Átadás 2-3 fában.



13. ábra Összevonás 2-3 fában.

2.6.4 A Fájlménü



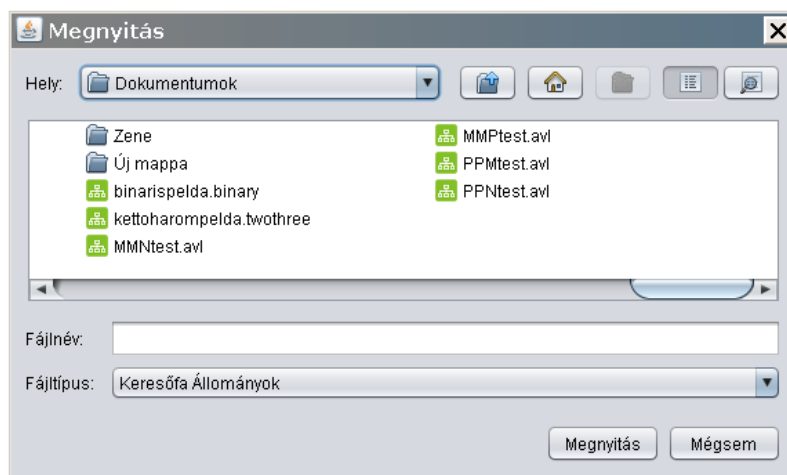
14. ábra Fájl menü.

A Fájl menüben található az „Új”, „Megnyitás” és „Mentés” menüpontok.

2.6.4.1 Az Új menüpont

Az új menüpont alatt érhetjük el a három fa létrehozásáért felelős menüpontokat.

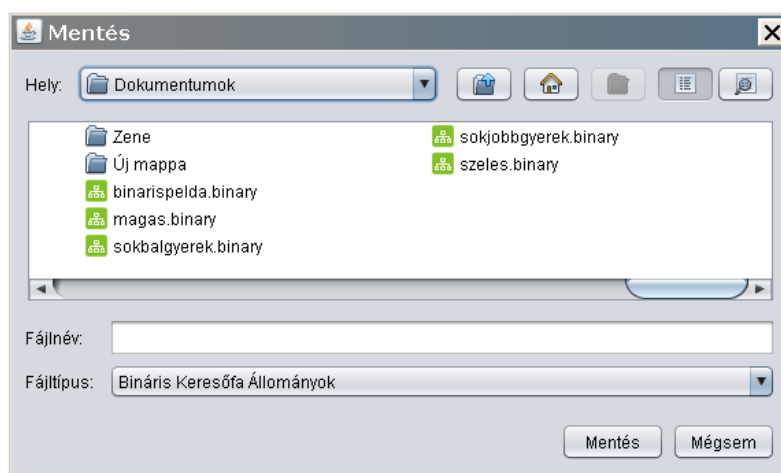
2.6.4.2 A Megnyitás menüpont



15. ábra Megnyitás dialógus ablak.

A Megnyitás menüpont aktiválása után a fájlrendszerben kiválaszthatunk a kívánt keresőfa állományt, majd betölthetjük a Megnyitás gombra kattintva.

2.6.4.3 A Mentés menüpont



16. ábra Mentés dialógus ablak.

A mentés menüpont aktiválása után elmenthetjük az aktuális fájlt, a kiválasztott helyre a kiválasztott néven.

2.6.5 A Szerkesztés menü



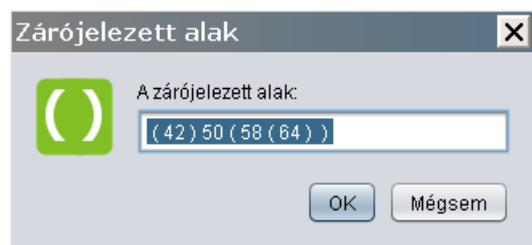
17. ábra Szerkesztés menü.

A Szerkesztés menüben érhetjük el a „Feltöltés véletlen értékekkel” és a „Zárójelezett alak előállítása” menüpontokat.

2.6.5.1 Feltöltés véletlen értékekkel

Beszúr az aktuális fába 10 darab 0 és 200 közötti véletlenszerűen kiválasztott értéket egymás után.

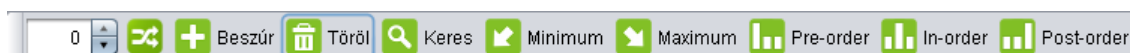
2.6.5.2 Zárójelezett alak előállítása



A menüpontra kattintva az alábbi üzenetablak ugrik fel. Itt láthatjuk az előállított zárójelezett alakot.

18. ábra Zárójelezett alakot megjelenítő panel.

2.6.6 A Műveletek eszköztár



19. ábra Műveletek eszköztár.

A műveletek eszköztár segítségével végezhetünk különböző műveleteket az aktuális fán.

2.6.6.1 Kulcs érték választás



21. ábra Kulcs érték választó mező



20. ábra Véletlen érték választó ikon

Az eszköztár bal oldalán található értékválasztó segítségével választhatjuk ki, milyen kulcs értékkel szeretnénk dolgozni. A bevitel történhet a mezőbe kattintás után billentyűzetről, vagy a mező melletti fel, le nyilak segítségével. A nyilak jobb oldalán található nyomógombra kattintva az értékválasztó mezőbe egy véletlen érték kerül.

2.6.6.2 Beszúr



22. ábra Beszúrás ikonja.

A beszúrás ikonra kattintva a kiválasztott kulcs értéket szúrhatjuk be a fába.

2.6.6.3 Töröl



23. ábra
Törlés ikonja.

A törlés ikonra kattintva a kiválasztott kulcs értéket törölhetjük a fából.

2.6.6.4 Keres



24. ábra
Keresés
ikonja.

A keresés ikonra kattintva a kiválasztott kulcs értéket kereshetjük meg a fában.

2.6.6.5 Minimum



25. ábra
Minimum
keresés
ikonja.

A minimum ikonra kattintva megkereshetjük az aktuális fában lévő minimális kulcs értéket.

2.6.6.6 Maximum



26. ábra
Maximum
keresés
ikonja.

A maximum ikonra kattintva megkereshetjük az aktuális fában lévő maximális kulcs-értéket.

2.6.6.7 Pre-order



27. ábra
Pre-order
bejárás
ikonja.

A „Pre-order” ikonra kattintva bejárhatjuk az aktuális fát pre-order bejárással. Pre-order bejárás során először meglátogatjuk a gyökeret, bejárjuk a bal részfat, majd bejárjuk a jobb részfat. [1]

2.6.6.8 In-order



28. ábra In-order bejárás ikonja.

Az „In-order” ikonra kattintva bejárhatjuk az aktuális fát in-order bejárással. In-order bejárás során először bejárjuk a bal részfat, meglátogatjuk a gyökeret, majd bejárjuk a jobb részfat. [1]

2.6.6.9 Post-order



29. ábra Post-order bejárás ikonja.

A „Post-order” ikonra kattintva bejárhatjuk az aktuális fát post-order bejárással. Post-order bejárás során először bejárjuk a bal részfat, bejárjuk a jobb részfat, majd meglátogatjuk a gyökeret. [1]

Megjegyzés: az imént említett nevezetes bejáró algoritmusok értelemszerűen csak bináris fákra indíthatók.

2.6.7 Az Animáció eszköztár



30. ábra Animáció eszköztár.

Az Animáció eszköztár a felület jobb szélső oldalán található. Segítségével az aktuális animációt befolyásolhatjuk.

2.6.7.1 Következő lépés



31. ábra
Következő
lépés ikonja

A következő lépés ikonjára kattintva az aktuális animációt léptethetjük tovább.

2.6.7.2 Folyamatos lejátszás



32. ábra
Folyamatos
lejátszás
bekapcsolva



33. ábra
Folyamatos
lejátszás
kikapcsolva

A folyamatos lejátszás ikonjára kattintva szabályozhatjuk, hogy az aktuális animáció folyamatosan haladjon tovább. Ne legyen szükség a „Következő lépés” ikonjára kattintani. Az ikonra való újbóli kattintás kikapcsolja ezt a lehetőséget.

2.6.7.3 Ugrás



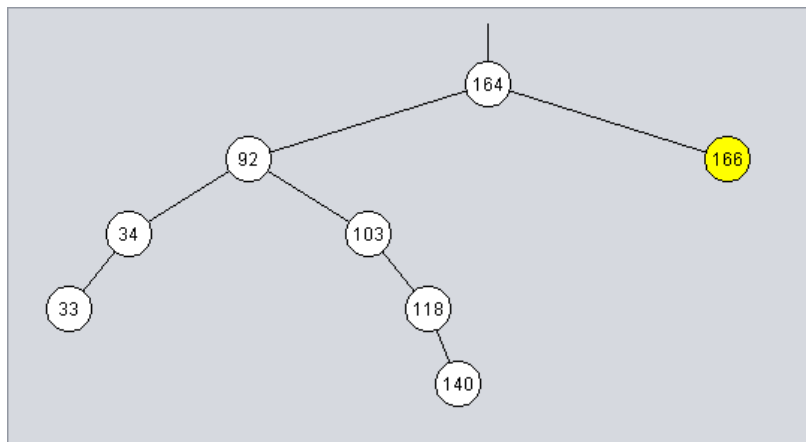
Az ugrás ikonjára kattintva az aktuális animációt gyorsan befejezhetjük. Nem kell megvárni, amíg az véget ér.

2.6.7.4 Megjelenés

A megjelenés ikonjára kattintva választhatjuk ki, hogy az aktuális fa szoros vagy szimmetrikus elrendezésű legyen.



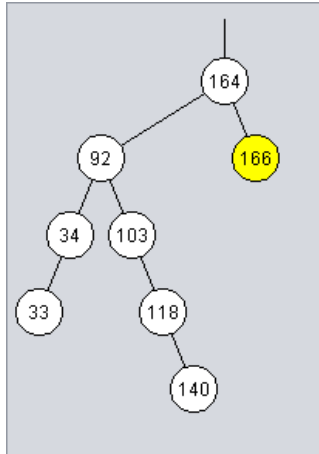
34. ábra
Szimmetrikus
megjelenés
ikonja



35. ábra Szimmetrikus elrendezésű bináris fa



36. ábra
Szoros
megjelenés
ikonja

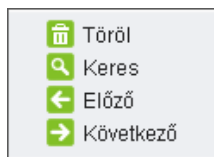


37. ábra Szoros elrendezésű bináris fa

2.6.8 Helyi menük

A grafikus felületen a fa egy megfelelő csúcsára jobb egérgombbal kattintva előhozhatjuk a helyi menüt.

2.6.8.1 Bináris és AVL keresőfa helyi menüje



38. ábra Bináris és
AVL-fa helyi menüje.

A bináris vagy AVL fa egy csúcsára kattintva előhozhatjuk a helyi menüt.

A törlés és keresés menüpontra kattintva az aktuálisan kijelölt csúcsot törölhetjük, illetve megkereshetjük.

2.6.8.1.1 Előző



39. ábra
Előző kulcs-
érték ikonja.

Az előző menüpontra kattintva megkereshetjük az aktuálisan kijelölt csúcs előzőjét.

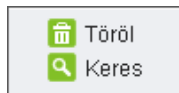
2.6.8.1.2 Következő



40. ábra
Következő
kulcs-érték
ikonja.

A következő menüpontra kattintva megkereshetjük az aktuálisan kijelölt csúcs következőjét.

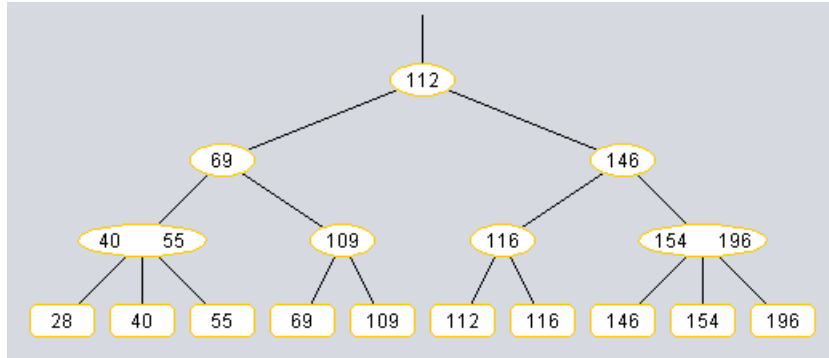
2.6.8.2 2-3 fa helyi menüje



A 2-3 fa helyi menüjét csak leveleken aktiválhatjuk. Az aktuálisan kijelölt levelet törölhetjük, vagy kereshetjük a megfelelő menüpontra kattintva.

41. ábra 2-3 fa helyi menüje.

2.6.9 Animáció rajzolási területe



42. ábra Animáció rajzolási területe.

A felület közepén található. Itt láthatjuk az aktuális fát grafikusán ábrázolva. Szintén itt láthatjuk az egyes animációs lépéseket.

2.6.10 Algoritmus megjegyzés területe



43. ábra Algoritmus megjegyzés területe.

A felület legalsó részén található. Benne az aktuális animációs lépés megjegyzéseit olvashatjuk.