

11. hét: bináris fák

Czirkos Zoltán · 2023.11.08.
Bináris fák – fák építése, mélységi bejárás rekurzióval.

Tartalom

- Negyedik kis ZH
- Felkészülés a laborra:
- Bináris fák: a keretprogram
- Fa kiírása
- Elemek száma és összege
- Elem megkeresése
- Fa építése
- A fa magassága
- Negálás
- Tükrözés
- További feladatok

1. Negyedik kis ZH

A hétfői alkalmakon lesz a negyedik kis ZH.
Este pedig a pótzárthelyi...

2. Felkészülés a laborra:

- A rekurzióról szóló előadás áismétlése.
- A bináris fákról tanultak áttekintése.

3. Bináris fák: a keretprogram

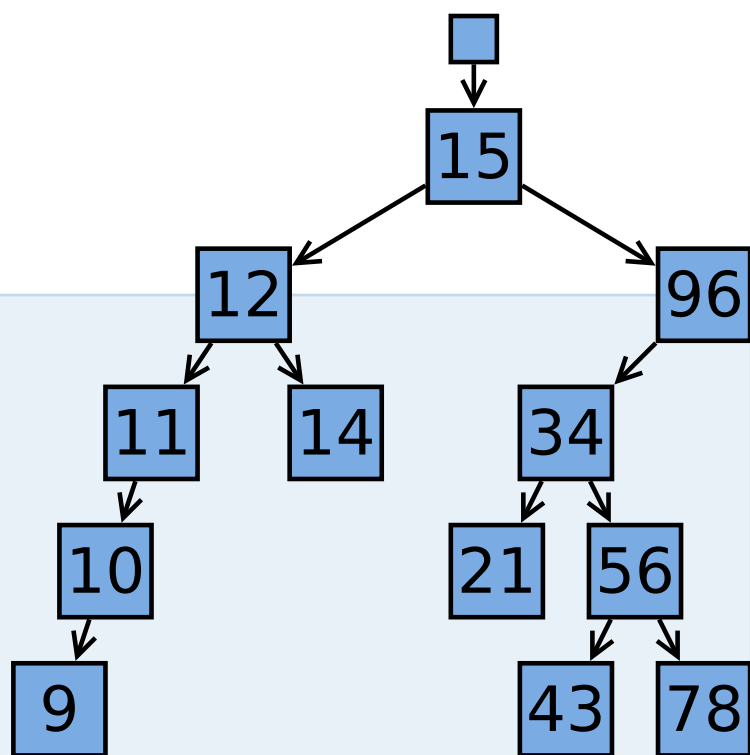
Az alábbi program létrehoz egy bináris keresőfát. A fa elemei egy egész típusú értéket tartalmaznak. Az órai feladatokat a program által létrehozott fával tudjátok kipróbálni. A létrehozott fa jobb oldalt látható.

```
class BinFa:
    def __init__(self, ertekek):
        self.ertekek = ertekek
        self.bal = None
        self.jobb = None

    def beszur(gyoker, ertekek):
        if gyoker is None:
            gyoker = BinFa(ertekek)
        elif ertekek < gyoker.ertekek:
            gyoker.bal = beszur(gyoker.bal, ertekek)
        elif ertekek > gyoker.ertekek:
            gyoker.jobb = beszur(gyoker.jobb, ertekek)
        else:
            pass
        return gyoker

    def main():
        tesztadat = [15, 96, 34, 12, 14, 56, 21, 11, 10, 9, 78, 43]
        gyoker = None
        for x in tesztadat:
            gyoker = beszur(gyoker, x)

        main()
```



4. Fa kiírása

Írj rekurzív függvényt, amely inorder (bal-gyökér-jobb) bejárja a fát, és kiírja a tárolt elemeket. A számokat növekvő sorrendben kell megkapjad, mert keresőfáról van szó.

5. Elemek száma és összege

Írj rekurzív függvényt, amely megszámolja és visszaadja a fa elemeinek számát! Ellenőrizd az algoritmus által adott eredményt a rajz alapján!

Írj rekurzív függvényt, amely meghatározza a fában tárolt számok összegét! Ellenőrizd ezt is a rajz alapján (vagy a listában tárolt számok alapján)!

6. Elem megkeresése

Írj függvényt, amely megkeres egy elemet a fában, és visszaadja a megtalált csomópont referenciáját! A visszatérési érték legyen **None**, ha az adott szám a fában nem szerepel.

► Tipp

Ezt a függvényt iteratíván is könnyedén meg lehet csinálni, mert nem kell elágaznia a keresésnek. Választásod szerint adj iteratív vagy rekurzív megoldást!

7. Fa építése

Töröld most ki a fát építő függvényt és a fa csomópontját definiáló osztályt! A feladatod az lesz, hogy megírd ezeket egyedül újra.

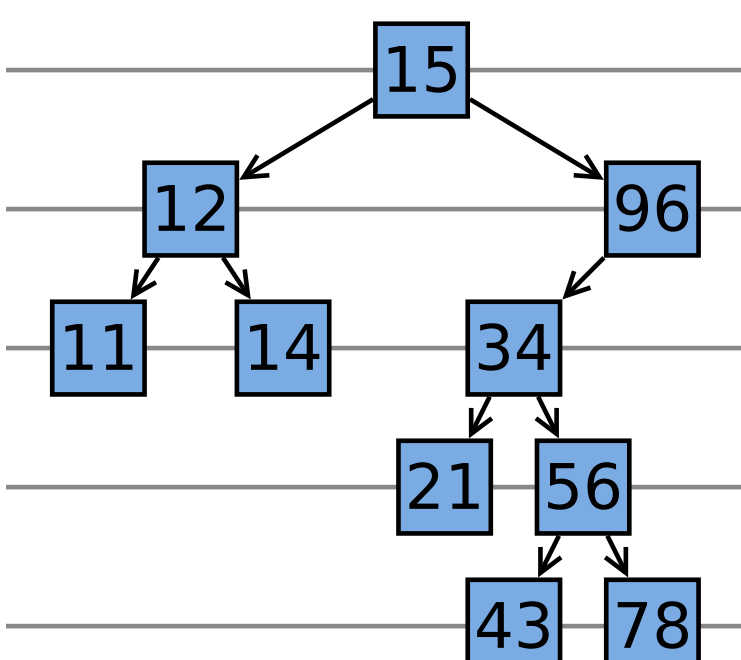
► Tipp az osztályhoz

► Tipp az építéshez

8. A fa magassága

Milyen magas a fa? Írj rekurzív, globális változót nem használó függvényt a magasság meghatározására!

► Tipp



9. Negálás

Írj függvényt, amely ellentettjére változtat, azaz -1-szeresére szoroz minden elemet a fában!

Keress most meg egy elemet a fentebb megírt keresőfüggvényeddel. Mit tapasztalsz? Miért történik ez? Hogyan módosítanád a kereső függvényt, hogy működjön az így kapott fán? (Ha kell, rajzold le egy kis részletét a gyökértől indulva a negált fának, és képzeletben hajtsd rajta végre az algoritmust! Ki is írathatod a negált fát az inorder bejárás függvényeddel.)

10. Tükrözés

Írj egy rekurzív függvényt, amely tükröz egy paraméterként kapott fát!

► Tipp

Keresés a negált, tükrözött fában

Most működik a módosított kereső függvény? Miért? Írasd ki a fa tartalmát az inorder függvénnyel, vagy készíts rajzot!

11. További feladatok

Ha végeztél, dolgozhatsz a nagy házi feladatodon vagy a kiadott szorgalmi feladatokon is.

