

G01. Rendezés bináris fával. Mélységi bejárás*

Gráfok

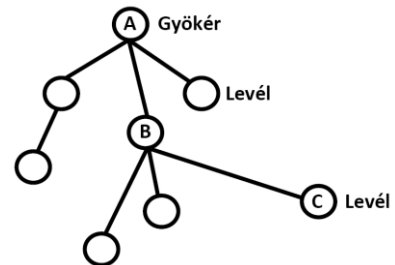
Összetett adatszerkezet leírására jól használható matematikai segédeszköz a gráf. **Ha csúcsokat éllel kötünk össze, gráfot kapunk.** Gráffal szemléltethetjük pl. egy ország úthálózatát, vagy egy meghajtó mappaszerkezetét.

Az egymáshoz csatlakozó éllel sorozatát **útnak** nevezzük. **A fa olyan gráf, amelyben bármely két csúcsot pontosan egy út köt össze** (nincs benne kör).

A fát általában egy kitüntetett csúcsból indulva rajzoljuk fel, ez a csúcs a fa **gyökere**. A fát már meglévő csúcsaiból élként „növesztjük”, a kapott új csúcsokat a kiindulási csúcs **leszármazottjainak** nevezzük. Ábránkon például A leszármazottja B, B leszármazottja C. A C csúcsnak már nincs több leszármazottja, az ilyen csúcsot a fa **levelének** hívjuk.

Az ábrán a B csúcs egy újabb fa gyökere, ezt az eredeti fa **részfájának** nevezzük.

A programozásban gyakran használjuk a bináris fát. **A bináris fában minden csúcsnak legfeljebb két leszármazottja lehet.**



Rendezés bináris fával

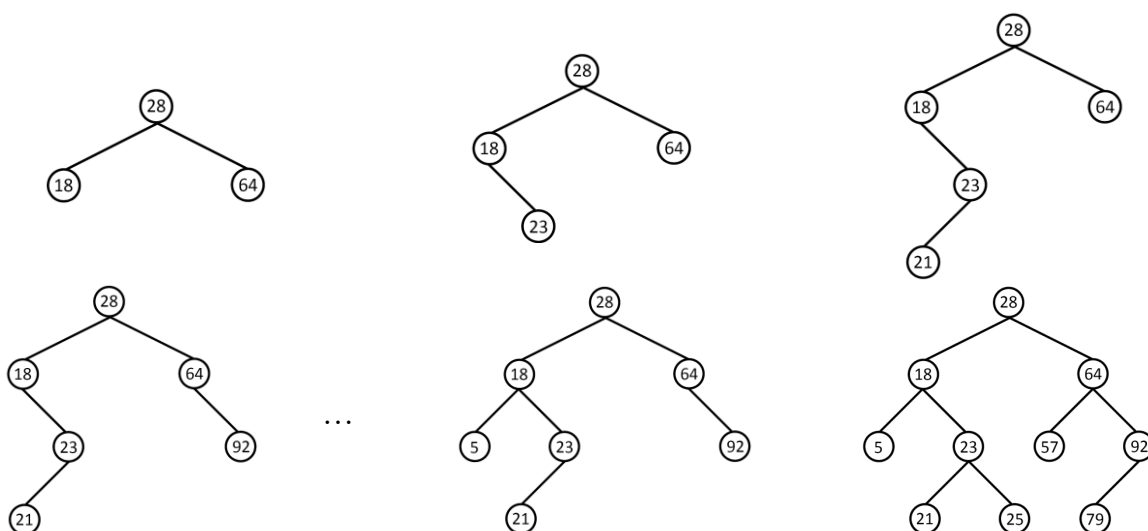
Bináris fa segítségével meglehetősen gyors rendezőalgoritmust készíthetünk. A módszer lényege a következő:

- A rendezendő sorozat első tagját a fa gyökerébe tesszük.
- Ha a következő tag kisebb a gyökérélemnél, akkor azt a bal oldali rész fába tesszük, egyébként a jobb oldaliba.
- Ha az adott helyen már van elem, akkor az lesz a fa új gyökere, és az algoritmust az előző ponttól folytatjuk.

Az ábrán a rendezőfa felépítésének kezdő és befejező lépéseit látjuk a

28, 18, 64, 23, 21, 92, 5, 25, 57, 79

tagokból álló sorozat esetén:

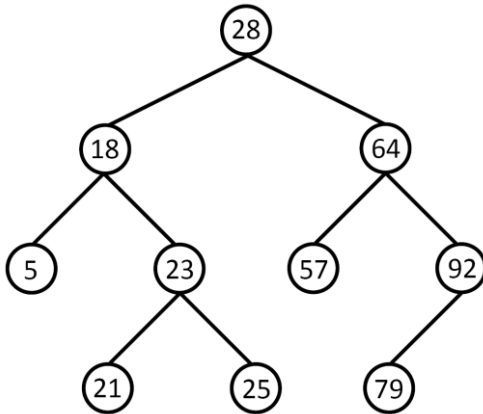


* Forrás: Farkas Csaba: A programozás alapjai Visual Basic .NET-ben, Jedlik Oktatási Stúdió, 2009 p. 206-211

Az eljárás algoritmus:

A rendezendő tömb elemeit az az n elemű x vektor tartalmazza. A tömb elemeinek számozása 1-től indul, így a gyökérbe az $x[1]$ kerül, gyökér indexe 1.

A bal oldali leszármazottakat a **bal**, a jobb oldaliakat **jobb** tömb tartalmazza. Példánkban a gyökér bal oldali leszármazottja a 2. indexű elem: $\text{bal}[1]=2$, jobb oldali leszármazottja pedig a 3. elem: $\text{jobb}[1]=3$. A bal és jobb tömbben **0** érték jelzi, ha nincs bal vagy jobb oldali leszármazott.



i	x(i)	bal(i)	jobb(i)
0	28	2	3
1	18	7	4
2	64	9	6
3	23	5	8
4	21	-1	-1
5	92	10	-1
6	5	-1	-1
7	25	-1	-1
8	57	-1	-1
9	79	-1	-1

Az algoritmus érdemi része a következő.

i : az elhelyezendő elem indexe
 l : sikerült-e már elhelyezni
 gy : a vizsgált (rész)fa gyökérelemének indexe

```
Az 1. elem a gyökérbe került, így 2.-től kezdünk
Ciklus i = 2-től n-ig
  //Mindig a gyökérből indulunk
  gy = 1
  l = false
  Ciklus amíg l = hamis
    Ha x[i] < x[gy]
      //ha kisebb, akkor balra próbáljuk betenni
      Ha bal[gy] = 0
        bal[gy] = i
        l = igaz
      //ha ott már van elem, akkor az lesz az új gyökér
      Különben
        gy = bal[gy]
      Elágazás vége
    Különben
      //Ugyanez a jobb oldalra
      Ha jobb[gy] = 0
        jobb[gy] = i
        l = igaz
      Különben
        gy = jobb[gy]
      Elágazás vége
    Elágazás vége
  Ciklus vége
Ciklus vége
```

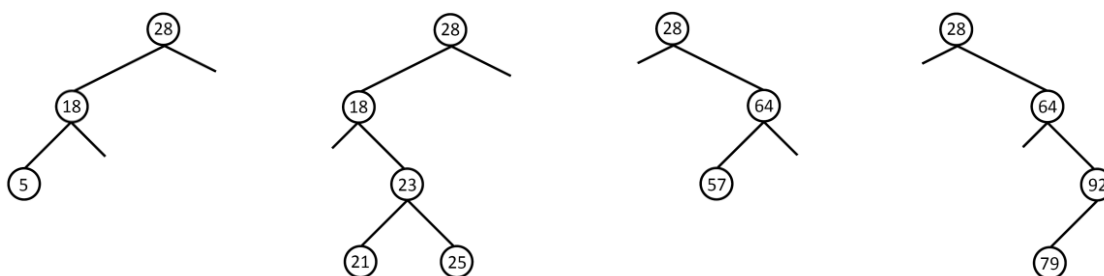
A rendezőfa bejárása (mélységi keresés)

A rendezőfáról a sorozat tagjait növekvő sorrendben a fa bejárásával olvashatjuk le. Mivel a gyökérelemnél kisebb elemek a gyökérelemtől mindig balra, a nagyobbak pedig jobbra kerültek, így a fa bejárásának algoritmus a következő (a fa bejárásának ez a módja az ún. *mélységi bejárás* egyik változata):

- Bejárjuk a bal oldali részfat.
- Kiírjuk a gyökérelemet.
- Bejárjuk a jobb oldali részfat.

Az eljárás önmagát hívja, tehát egyszerűen megvalósíthatjuk egy rekurzív függvény segítségével. Példánkban a rendezőfa bal oldali részfája maga is egy bináris fa, így bejárását annak bal oldali részével kezdjük, a „legmélyebb” ponttal: **5**. Ezt követi a bal oldali részfa gyökere: **5, 18**, majd bejárjuk a bal oldali részfa jobb oldali részfáját: **5, 18, 21, 23, 25** (ld. ábra).

Kiírjuk a rendezőfa gyökerét: **5, 18, 21, 23, 25, 28**.



A jobb oldali részfa szintén egy bináris fa. Ennek bejárása hasonlóan történik: **28, 57, 64, 79, 92** (ld. ábra).

A program kódja C#-ban:

```
namespace RendezésBinárisFával
{
    internal class Program
    {
        //A rendezendő tömb
        //Ezúttal a tömb elemeit 1-től indexeljük
        static int n = 20;
        static int[] x = new int[n+1];
        //A bal és jobb tömbben 0 azt jelzi, hogy "üres" az adott hely
        static int[] bal = new int[n+1];
        static int[] jobb = new int[n+1];

        static void Main(string[] args)
        {
            //A tömb feltöltése véletlenszerűen
            Random rnd = new Random();
            for (int i = 1; i <= n; i++)
                x[i] = rnd.Next(0, 100);

            Console.WriteLine("A rendezendő tömb:");
            for (int i = 1; i <= n; i++)
                Console.Write(x[i] + " ");
            Console.WriteLine();

            Rendezőfa();
            Console.WriteLine("A rendezett tömb:");
            Bejárás(1);
        }
    }
}
```

