

# A rendezési algoritmusok tanításának céljai és módszerei

Bernát Péter

bernatp@inf.elte.hu  
ELTE INFORMATIKA DOKTORI ISKOLA

**Absztrakt.** Cikkemben a rendezési algoritmusok tanításának lehetséges módszereit és céljait járom körül. Rendszerezem és bemutatom az elsősorban szemléltetésre vagy kísérletezésre épülő módszereket, és azokat az interneten elérhető magyar illetve külföldi oktatási segédanyagokkal, valamint saját ötletekkel illusztrálom.

*„Hallom és elfelejtem, látom és emlékszem rá,  
csinálok és megértem”  
(kínai közmondás)*

## 1. Bevezetés

A rendezési algoritmusok témaköre a programozástanításon belül hálás téma: a rendezés a hétköznapokból is jól ismert, könnyen szemléltethető probléma, amelyre számos hatékony vagy kevésbé hatékony megoldás adható.

Az alsó tagozatban az informatikaoktatás a rendezési algoritmusok végrehajtására, a középiskolában azok számítógépes megvalósítására koncentrálnak. Rövid számbavételük után rendszerezem és bemutatom tanításuk lehetséges módszereit, illetve rámutatok a velük teljesíthető tanítási célokra. Az egyes módszereket az interneten elérhető oktatási segédanyagokkal, illetve saját programokkal illusztrálom.

## 2. A rendezési algoritmusok

Az oktatásban a hangsúlyt a helyben rendező algoritmusokra szokták helyezni: a nem hatékonyak közül az egyszerű cserésre, a minimumkiválasztásosra, a beillesztésesre és a buborékosra, a hatékonyak közül például a quick sortra vagy az összefuttatásos rendezésre.

### 2.1 Egyszerű cserés rendezés

Ha az egyszerű cserés rendezés egyáltalán előfordul a tanításban, elsőként szokták bemutatni. Leginkább a minimumkiválasztásos módszer előzményének tekinthető, amely nemcsak hatékonyabb, de véleményem szerint a hétköznapi gondolkodáshoz is közelebb álló.

### 2.2 Minimumkiválasztásos rendezés

A minimumkiválasztásos rendezési elv természetes ötletre épül. Valószínűleg sokan így rendeznek dolgokat anélkül, hogy ismernék a módszer elnevezését. Általános sorozatok esetén a nem hatékony rendezési algoritmusok közül a leghatékonyabb: nagyságrendileg a többivel azonos mennyiségű összehasonlítást végez, viszont a mozgások száma kevesebb.

### 2.3 Beillesztéses rendezés

A beillesztéses módszer a hétköznapokban ugyancsak gyakran előfordul: így szoktuk például elrendezni a kezünkbe vett játékkártyalapokat. Habár a minimumkiválasztásos módszernél általános esetben kevésbé hatékony, előnye, hogy nem kell az összes adatnak rendelkezésre állnia a rendezés megkezdésekor, illetve hogy a majdnem rendezett sorozatok esetén jobban teljesít a minimumkiválasztásosnál.

### 2.4 Buborékos rendezés

Népszerű nem hatékony rendezési algoritmus. Az alapötlete, azaz hogy a sorozat folyamatosan rendeződjön, reményre adhat okot, mégis a minimumkiválasztásos módszernél hatékonytalanabb általános esetben. Ráadásul az alapötlet nem mondható éppen természetesen adódónak, ezért egyesek egyenesen számúznák az oktatásból [1]. Megjegyzendő azonban, hogy a majdnem rendezett sorozatok esetén ez a módszer is hatékonyabb a minimumkiválasztásosnál.

### 2.5 Quick sort rendezés és összefuttatásos rendezés

A quick sort és az összefuttatásos rendezés hatékonyabb az előzőeknél, és egyforma hatékonyságúak. Mindkettő rekurzóra épül, így az „oszd meg és uralkodj” elvű algoritmusok szép példái.

### 2.6 Szétosztó rendezés

Esetenként speciális feltételekre épülő módszerek is szóba kerülnek a tanítás során, így például a szétosztó rendezés. A rendezendő sorozat elemeinek 1 és  $N$  közötti, egymástól különböző kulcsértékekkel kell rendelkezniük. Nem kell összehasonlításokat végezni, viszont szükség van egy az eredetivel megegyező méretű új adatszerkezetre (tömbre, listára).

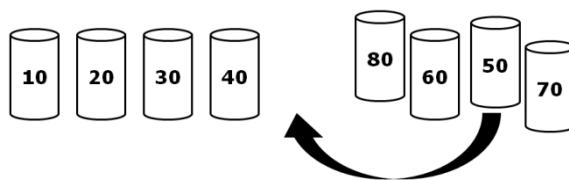
## 3. A rendezési algoritmusok végrehajtásának tanítása

Az alsó tagozatos diákok nemcsak használni kezdik a számítógépet, de a működésükkel kapcsolatos alapismeretekre is szert tesznek. Számítógéphasználat nélkül, játékos formában ismerkedhetnek meg például a bináris számokkal, a kép és a szöveg egyszerű kódolási lehetőségeivel, valamint kereső- és rendezőalgoritmusokkal [1].

### 3.1. Szemléltetés és kísérletezés tárgyakkal

Esetükben a rendezési probléma felvetésére, és néhány algoritmus megtanítására a tárgyakkal kísérletezés a legalkalmasabb. A tárgyak sorba rendezése valamilyen szempont szerint a hétköznapai valósághoz közelálló, és a cselekvésen keresztüli tanulás különösen ebben az életkorban nagyon hatékony.

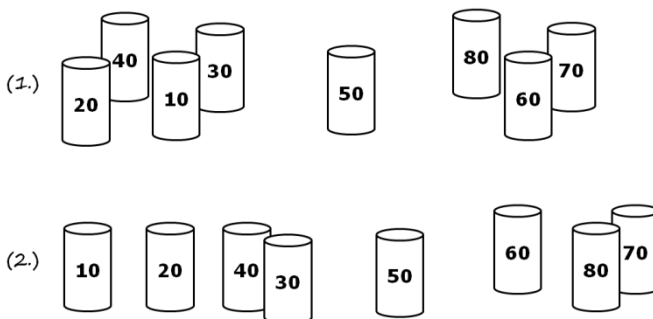
Olyan tárgyakat érdemes választani, amelyek valamilyen nem látható tulajdonságuk, célszerűen a tömegük alapján egyértelműen sorrendbe állíthatók. Mivel két elem összehasonlítása csak mérleggel lesz lehetséges, a rendezés érzékelhetően elempárok összehasonlításán fog alapulni. Tárgyaknak megfelelők a különböző mennyiségű pénzérmével feltöltött egyforma kapszulák, kétkarú mérlegek pedig az iskola kémiaszertárából kölcsönözhetőek.



1. ábra: Súlyok rendezése a minimumkiválasztásos módszerrel [2]

Egy lehetséges terv a következő [2]: a diákok négyfős csoportonként 8 tárgyat és egy kétkarú mérleget kapjanak. Rávezetésként határozzák meg 2 súly közül a nehezebbet, majd az összes közül a legnehezebbet. Ezután rendezzenek súlyuk szerint 3 elemet, majd pedig (tetszőleges módszerrel) az összeset.

A rávezető feladatok után megtanítható egy nem hatékony és egy hatékony rendezési algoritmus. Választhatjuk például a minimumkiválasztásost (1. ábra) és a quick sortot (2. ábra). A tanulók megszámoľhatják a szükséges összehasonlításokat (mérésszám) az első, illetve a második esetben, és megállapíthatják, hogy különböző hatékonyságú rendezési módszerek léteznek. Érdeklődő osztályokban további algoritmusok is bemutatathatók, így például a beillesztéses vagy az összefuttatásos rendezés.



2. ábra: Súlyok rendezése a quick sort algoritmussal [2]

### 3.2. Tanítási célok teljesítése

A foglalkozás során a diákok észrevétlenül kerülnek közelebb fontos fogalmakhoz: algoritmus, rendezés (adott objektumokat megfelelő tulajdonság kiválasztása után sorrendbe állíthatunk), hatékonyság (különböző lépésszámú megoldásokat lehet találni ugyanarra a problémára), rekurzió („oszd meg és uralkodj!”). A próbálkozások, illetve a bemutatott algoritmusok megértése, végrehajtása és összehasonlítása fejlesztik a gondolkodást. Mivel pedig a feladatokat csoportosan kell végrehajtani, a tanulók szociális készségei is finomodnak.

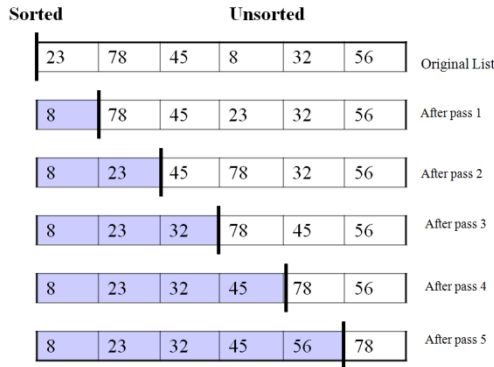
## 4. A rendezési algoritmusok számítógépes megvalósításának tanítása

A rendezési probléma a sorozathoz sorozatot rendelő feladattípusba tartozik, és általában a programozási tételekhez kapcsolódóan jelenik meg az oktatásban. A tanulóknak többféle rendezési algoritmust kell megérteniük, leírnyelvvel leírniuk, kódolniuk, hatékonysági szempontból megvizsgálniuk, és természetesen felhasználniuk különböző feladatok megoldásában. Ezekben a lépésekben is különböző személtető-kísérletező módszerek segíthetnek.

### 4.1. Szemléltetés képpel

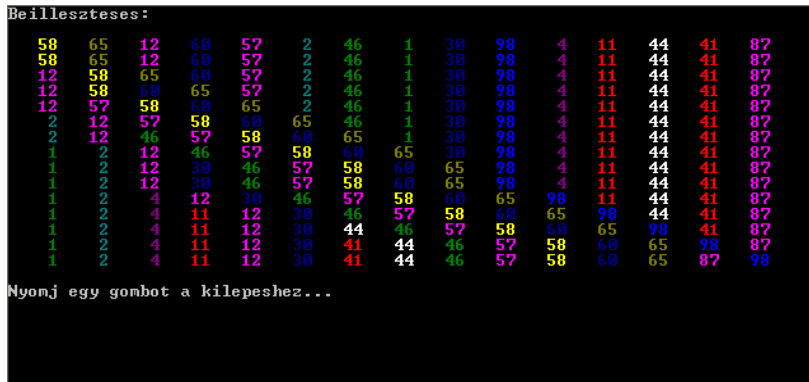
Az érintett korosztályban elegendő képpel vagy animációval szemléltetni, bár elképzelhető, hogy egyes tanulókon a tárgyakhoz visszatérés segíthet.

A képes szemléltetések a rendezési módszerek megértésében nyújthatnak segítséget. Ezek jellemzően a rendezőalgoritmusok külső ciklusainak lépéseivel ábrázolják az adatsorozatot. A „nagy mozgások” bemutatásával rávilágítanak az algoritmusok stratégiájára, rejtve maradnak viszont az elemi lépések (3. ábra).



3. ábra: A minimumkiválasztásos rendezés szemléltetése képpel [3]

Ilyen képek program segítségével is generálhatók, így véletlenszerű, vagy speciális sorozatok mozgása is megfigyelhető a rendezési algoritmusok közben. Érdeemes az elemekhez különböző színeket hozzárendelni a kiíráskor (például maradékszámítással), hogy a helyváltogatások és helyben maradások látványosak legyenek (4. ábra).

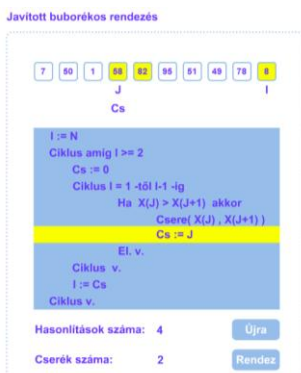


4. ábra: A beillesztéses rendezés szemléltetése programmal (saját program)

### 4.2. Szemléltetés animációval

Az animációk a rendezési algoritmusok megértésétől a leírason és a kódoláson át a hatékonyságvizsgálatig mindegyik lépésben használhatók. Ezek általában az algoritmusok belső ciklusainak lépéseivel ábrázolják a sorozatot. Bemutatják tehát az algoritmusok teljes működését, viszont nehezebben vehető bennük észre a módszerek lényege.

Az algoritmus megértésében segít, ha a lejátszás sebessége változtatható (esetleg visszafordítható) [4], mert így könnyebb a szemléltetést követni, illetve tanárként magyarázatokat hozzáfűzni. Ha a sebesség nem is változtatható, szüneteltetni minden animációt lehet.



5. ábra: A javított buborékos rendezés szemléltetése animációval [5]

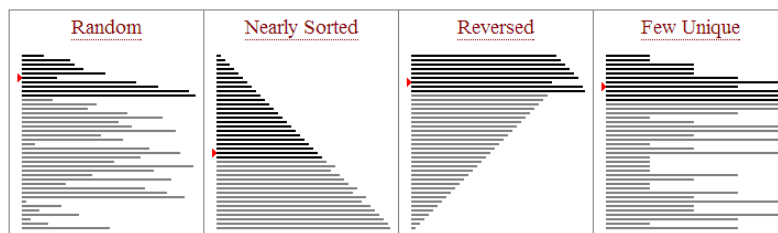
Az algoritmusok leírását és kódolását megkönnyítik a ciklusok mutatóinak „helyét” jelölő animációk [4][5][6][7] (5. ábra). Egyes bemutatókhoz kódot is mellékelnek, amelyen kiemelik az aktuálisan végrehajtott utasítást. Ezt jómagam nem tartom szerencsésnek, hiszen az önálló kódolás egyben a megértés próbája is.

## Insertion Sort



Problem Size: [20](#) · [30](#) · [40](#) · [50](#) Magnification: [1x](#) · [2x](#) · [3x](#)

Algorithm: [Insertion](#) · [Selection](#) · [Bubble](#) · [Shell](#) · [Merge](#) · [Heap](#) · [Quick](#) · [Quick3](#)



6. ábra: A beillesztéses rendezés szemléltetése speciális sorozatokkal [8]

Végül, a hatékonyságvizsgálatot a speciális sorozatokat is rendező animációk támogathatják, amelyek láthatóvá teszik a sorozat típusától függő viselkedés- és hatékonyságbeli különbségeket adott rendezési algoritmusnál (6. ábra). Az animációk továbbá számolhatják és megjeleníthetik a rendezés során végzett összehasonlítások, illetve mozgatások számát [5]. Különleges esetnek tekinthető, amikor egy program nem mutat animációt, de az említett értékeket meghatározza különböző sorozatok, illetve rendezési algoritmusok esetén (7. ábra).

```

A sorozat: 29 22 43 47 40 28 35 12 36 15
A sorozat tipusa: veletlen

Hasonlítások      Masolasok      Osszesen
Egyszeru cseres:  45              84            129
Minimumkivalasztasos: 45              27            72
Buborekos:        45              84            129
Javitott buborekos: 45              84            129
Beilleszteses:   35              84            119
Javitott beilleszteses: 35              46            81

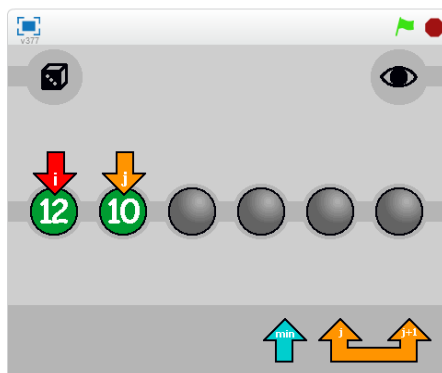
Ujra: Enter, kilepes: Esc...

```

7. ábra: Különböző algoritmusok hatékonysága különböző sorozatok esetén (saját program)

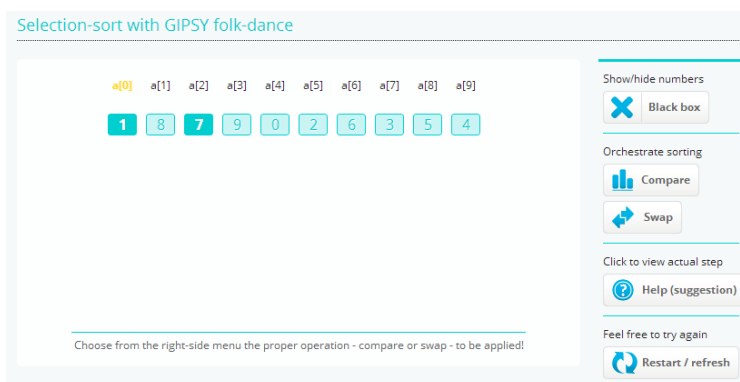
### 4.3. Kísérletezés szimulációs programmal

A szimulációs programok az algoritmusok biztos megértésében segíthetnek. Habár a tárgyakkal kísérletezéshez képest kevesebb mozgás- és érzékszervvel vannak kapcsolatban, különböző szolgáltatásokat nyújthatnak (8. ábra).



8. ábra: Rendezés szimulációval (saját program) [9]

A szolgáltatások közül a legfontosabb, hogy ellenőrizhetik az aktuálisan végrehajtott lépés helyességét a kiválasztott rendezési algoritmustól függően, sőt, folyamatos instruálással akár meg is taníthatják az algoritmust (9. ábra).



9. ábra: Rendezés szimulációban, ellenőrzési lehetőséggel [6]

### 4.4. Tanítási célok teljesítése

A rendezési algoritmusok számítógépes megvalósítása valamilyen adatszerkezet (tömb, lista) célszerű használatát, egymásba ágyazott ciklusok szervezését, illetve a konkrét algoritmustól függően rekurzív eljárás készítését igényelheti. Számításokat lehet végezni a futási idővel és a tárhellyel kapcsolatban.

A szóban forgó algoritmusok a programozási tételek közül a legösszetettebbek. Leírásuk és kódolásuk a gondolkodást próbára tévő feladat, amelynek megoldását a bemutatott módszerekkel képességektől és előismeretektől függően lehet kisebb vagy nagyobb mértékben támogatni.

Nem utolsó sorban a rendezési algoritmusok felhasználhatók különböző matematikai, természettudományos, nyelvi és egyéb feladatok megoldásában, vagyis a témakör megtanításával a programozás eszközként felhasználását is támogathatjuk a többi műveltségi területen.

## Irodalom

1. Owen Astrachan: *Bubble Sort: An Archaeological Algorithmic Analysis*  
<http://www.cs.duke.edu/~ola/papers/bubble.pdf> (hozzáférés dátuma: 2013.11.03.)
2. T. Bell, I. H. Witten, M. Fellows: *Computer Science Unplugged, An enrichment and extension programme for primary-aged children*  
[http://csunplugged.org/sites/default/files/activity\\_pdfs\\_full/unpluggedTeachersMar2010-USletter.pdf](http://csunplugged.org/sites/default/files/activity_pdfs_full/unpluggedTeachersMar2010-USletter.pdf) (hozzáférés dátuma: 2013.11.03.)
3. *Sorting Algorithms Section 3, C.Eng 213 Data Structures*  
[http://cow.ceng.metu.edu.tr/Courses/download\\_courseFile.php?id=5451](http://cow.ceng.metu.edu.tr/Courses/download_courseFile.php?id=5451) (hozzáférés dátuma: 2013.11.03.)
4. *Sort Animation*  
<http://www.cs.oswego.edu/~mohammad/classes/csc241/samples/sort/Sort2-E.html> (hozzáférés dátuma: 2013.11.03.)
5. *Informatikai oktatóprogramok, Oktatóprogramok készítése flash-ben*  
<http://tenger.web.elte.hu/flash/index.htm> (hozzáférés dátuma: 2013.11.03.)
6. *Algo-rhythmics*  
<http://algo-rhythmics.ms.sapientia.ro> (hozzáférés dátuma: 2013.11.03.)

7. *Programozni egyszerű! Programozás alapjai kezdőknek, Top School*  
<http://www.topschool.hu/programozni-egyszeru.php> (hozzáférés dátuma: 2013.11.03.)
8. *Sorting Algorithm Animations*  
<http://www.sorting-algorithms.com/> (hozzáférés dátuma: 2013.11.03.)
9. Bernát P.: *Cserés rendezések*  
<http://scratch.mit.edu/projects/87514/> (hozzáférés dátuma: 2013.11.03.)