

A rendezési algoritmusok tanításának módszerei és céljai a közoktatásban

REN DEZIRÉ

Absztrakt. A rendezési algoritmusok témaköre az informatikatanításon belül hálás téma: a rendezés fogalma a hétköznapiakból is jól ismert, algoritmizálási feladatként könnyen meghatározható és szemléltethető, amelyre a naivaktól a gyakorlatban használtakig számos megoldás adható. Cikkemben a rendezési algoritmusok általános és középiskolai tanításának lehetséges módszereit és céljait járom körül. Rendszerezem és az interneten elérhető példákkal illusztrálom a szemléltetésre és munkáltatásra épülő módszereket, rámutatok a módszerek használatával teljesíthető tanítási célokra, és végül tanulási szakaszonként kiválasztom a javasolható módszereket.

Kulcsszavak: általános és középiskolai tanítás; rendezési algoritmusok; szemléltető módszer; munkáltató módszer

1. Bevezetés

A rendezési algoritmusok témaköre az informatikatanításon belül hálás téma: a rendezés fogalma a hétköznapiakból is jól ismert, algoritmizálási feladatként könnyen meghatározható és szemléltethető, amelyre a naivaktól a gyakorlatban használtakig számos megoldás adható.

A közoktatásban jellemzően az összehasonlító rendezési algoritmusokkal foglalkoznak. Ezek minden lépésben két elemről döntenek el, hogy melyiküknek kell korábban előfordulnia a kimenetben. Közéjük kevésbé hatékony alapalgoritmusok, illetve a gyakorlatban inkább használt hatékony algoritmusok tartoznak. Alapalgoritmus például a kiválasztásos, a beillesztéses és a buborékrendezés, hatékonyabbak között a quicksort, az összefuttatásos rendezés, a Shell rendezés és a kupacrendezés.

Az iskolában a tárgyalt témakörrel két korosztály találkozik. Az alsó tagozatosok megtanulnak néhány rendezési algoritmust végrehajtani és hatékonysági szempontból összehasonlítani, a középiskolások az algoritmusok leírásával és számítógépes megvalósításával is foglalkoznak, és részletesebb hatékonysági vizsgálatot végeznek.

A rendezési algoritmusok megtanításának kézenfekvő módszerei a szemléltetés és a munkáltató módszer. A szemléltetés tárgyak, jelenségek, folyamatok

bemutatásának és elemzésének, a munkáltató módszer tevékenységek gyakorlásának egy-egy eszköze [1], az algoritmusok pedig tekinthetők tanulmányozandó folyamatnak, ha végrehajtójuk egy másik személy vagy a számítógép, illetve elsajátítandó cselekvésnek is, ha a végrehajtók mi magunk vagyunk.

A programozástanítás nemcsak programozási alapfogalmak átadásáról szól, hanem más tantárgyakhoz hasonlóan fejleszti a tanulók gondolkodását, és akár szociális készségeit is – a rendezési algoritmusok tanításával is elérhetők hasonló célok.

Cikkemben célom kategorizálni és az interneten elérhető példákkal illusztrálni a rendezési algoritmusok tanításának szemléltetésre és munkáltatásra épülő módszereit, rámutatni a módszerek használatával teljesíthető tanítási célokra, és végül tanulási szakaszonként kiválasztani a javasolható módszereket.

2. A rendezési algoritmusok végrehajtásának tanítása

Az alsó tagozatos diákok számítógéphasználat nélkül, játékos formában ismerkedhetnek meg a számítógépek működésével kapcsolatos alapismeretekkel, köztük a rendezési algoritmusokkal. Feladatuk megérteni, végrehajtani, és hatékonysági szempontból összehasonlítani néhány rendezési algoritmust.

2.1. Szemléltetés és munkáltatás tárgyakkal

Esetükben néhány algoritmus megtanítására a tárgyak használata a legalkalmasabb. Tárgyak sorba rendezése valamilyen szempont szerint a hétköznapi valósághoz közelálló, és a cselekvésen keresztüli tanulás különösen ebben az életkorban hatékony.

Érdekes a tömegük szerint sorba állítható tárgyakat választani. Mivel ezek összehasonlítása csak kétkarú mérleggel lehetséges, kellő figyelem irányul az összehasonlító rendezések alaplépésére, és könnyebben megszámlálhatók az elvégzett összehasonlítások. Tárgyaknak megfelelők például a különböző mennyiségű pénzérmével feltöltött egyforma kapszulák, mérlegek pedig az iskola kémiaszertárából is kölcsönözhetők. Ha a tárgyak alá sorszámokat ragasztunk, a felállított sorrend később könnyen ellenőrizhető lesz.

Egy lehetséges terv a következő. A diákok alkossanak négyfős csoportokat, kapjanak 8 tárgyat és egy mérleget. Mondjuk el, hogy egy lépésben mindig csak két tárgy súlyát szabad összehasonlítani. Rávezető feladatként határozzák meg két tárgy közül a nehezebbet, majd az összes közül a legnehezebbet. Beszéljük meg, hogy az

utóbbi feladat hogyan és hány lépésben oldható meg. Ezután rendezzenek a súlyuk szerint 3 tárgyat, majd próbálkozzanak meg az összes rendezésével.

A kísérletezéseket követően megtanítható egy alapalgoritmus és egy hatékony algoritmus. Választhatjuk például a kiválasztásos rendezést és a quicksortot. Ha a tanulók megszámozzák a szükséges összehasonlításokat az első, illetve a második esetben, megállapíthatják, hogy különböző hatékonyságú rendezési módszerek léteznek. Érdeklődés és rendelkezésre álló időkeret esetén további algoritmusok is bemutatathatók, így például a beillesztéses vagy az összefuttatásos rendezés.

2.2. Tanítási célok teljesítése

A foglalkozások során a diákok észrevétlenül kerülnek közelebb fontos fogalmakhoz: algoritmus, rendezés (adott dolgokat megfelelő tulajdonság szerint sorrendbe állíthatunk), hatékonyság (különböző lépésszámú megoldások létezhetnek ugyanarra a problémára) és rekurzió („oszd meg és uralkodj”). A próbálkozások, illetve a bemutatott algoritmusok megértése, végrehajtása és összehasonlítása fejlesztik a gondolkodást, mivel pedig a feladatokat csoportosan kell megoldani, a tanulók szociális készségei is fejlődhetnek.

3. A rendezési algoritmusok számítógépes megvalósításának tanítása

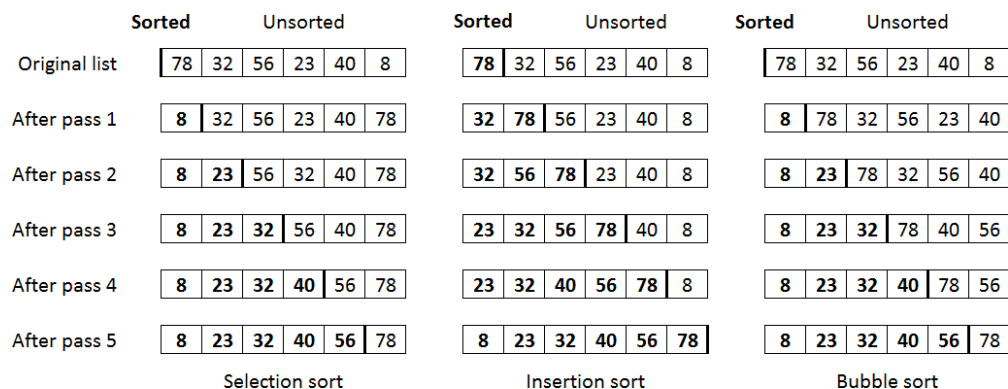
A rendezési probléma általában a programozási tételekhez kapcsolódóan jelenik meg a középiskolai programozásoktatásban. A tanulóknak többféle rendezési algoritmust kell megérteniük, leírniuk, kódolniuk, és hatékonysági szempontból megvizsgálniuk.

3.1. Szemléltetés képpel

A képes szemléltetések jellemzően a rendezési algoritmusok fő lépéseit mutatják be. Jól láthatóvá teszik a módszerek stratégiáit, viszont rejtve maradnak az elemi lépések.

Az alapalgoritmusokhoz készülő képek általában a külső ciklus lépéseivel ábrázolják a sorozatot (1. ábra). A kiválasztásos, a beillesztéses és a buborékrendezést közösen szemléltető ábrán megfigyelhető, hogy a módszerek a sorozatot egy rendezett és egy rendezetlen részre osztják fel, és minden lépésben egy elemmel bővítve a rendezett részt, végül a teljes sorozatot rendezetté teszik. Vizsgálható, hogy az elemek a rendezett részbe kerülés előtt, illetve után megváltoztatják-e a helyüket.

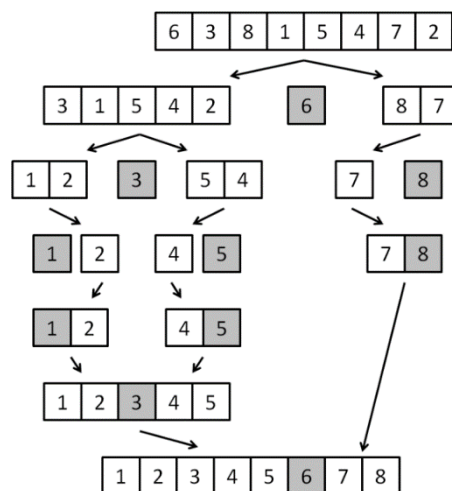
Ezekhez hasonló képek program segítségével is előállíthatók, így véletlenszerű, vagy speciális sorozatok rendezése is megtekinthető különböző algoritmusok esetén. Érdekes eltérő színeket hozzárendelni az elemekhez, hogy a mozgások és a helyben maradások látványosak legyenek.



1. ábra A kiválasztásos, a beillesztéses és a buborékredezés szemléltetése képpel [2]

A hatékony rekurzív rendezőalgoritmusokhoz készült képeken legtöbbször a rekurzió egyes szintjein ábrázolják a sorozatot [3]. Például a quicksorthoz készített ábrán megállapítható (2. ábra), hogy az algoritmus a kiinduló sorozatot egyre kisebbekre bontja, míg végül csak egyeleműek maradnak, majd a sorozatok többszöri egyesítésével előállítja a rendezett sorozatot. Hasonló szemléltető ábra készíthető az összefuttatásos rendezéshez is. Megfigyelhető, hogy a quicksort algoritmusban a részekre bontás az összetettebb, és az egyesítés az egyszerűbb, az összefuttatásos rendezés esetén viszont a könnyű részekre bontás ára a bonyolultabb egyesítés.

Ahhoz, hogy a módszerekben észrevegyük a rekurziót, a szemléltető képeket nem felülről lefelé, hanem „kívülről befelé” haladva kell értelmeznünk. Például az első lépés az utolsóval együtt alkotja a rekurzió legfelső szintjét: a kiinduló sorozatot valamilyen módon két nála kisebb sorozatra bontjuk, ezeket rendezzük, majd a két részt egyesítve megkapjuk a megoldást. Inentől kezdve rekurzívan gondolkozva a két részt külön-külön ugyanezzel a módszerrel rendezhetjük. A rekurzió legalsó szintjén az egyelemű részek állnak, amelyeket már nem kell rendezni.



2. ábra A quicksort rendezés szemléltetése képpel

3.2. Szemléltetés animációval

Az animációk két csoportra oszthatók. Az egyik csoportba tartozók a rendezési algoritmusok működését szemléltetik. Az összes elemi lépést bemutatják, viszont nehezebben vehető bennük észre a módszerek lényege.

Ezekben az animációkban az algoritmus működése jobban követhető, ha a lejátszás szüneteltethető, a sebessége megváltoztatható, esetleg visszafordítható, vagy ha az algoritmus által aktuálisan használt elemeket nyilakkal, színezéssel vagy más módon jelölik. Egyes interaktív animációk az algoritmus működését tetszőleges sorozaton be tudják mutatni, így velük különböző esetek szemléltethetők.

A Visualization of Data Structures and Algorithms weboldal célja egységes kezelőfelületű interaktív animációkkal szemléltetni gyakran használt adatszerkezeteket és algoritmusokat. A rendezések közül a kiválasztásos rendezéshez, a buborékrendezéshez, a quicksorthoz és az összefuttatásos rendezéshez tartalmaz animációt. Tetszőlegesen megadott sorozat rendezhető velük (az elemeket különböző magasságú oszlopok jelenítik meg), változtatható a lejátszás sebessége, sőt a lépések között szabadon lehet előre és hátra közlekedni egy csúszka segítségével.

Az animációk másik csoportja a rendezőalgoritmusok hatékonyságát szemlélteti párhuzamosan futó rendezések bemutatásával. Ezek vagy egyetlen rendezési algoritmus hatékonyságát mutatják be különböző típusú, vagy különböző algoritmusok két azonos típusú sorozatok esetén. Az algoritmusok működése közben össze is számolhatják az összehasonlításokat és az értékmásolásokat.

A Sorting Algorithm Animations weboldalon nyolcféle rendezési algoritmus hatékonyságát szemléltetik – a beillesztéses, a kiválasztásos, a buborék-, a Shell, az összefuttatásos és a kupacrendezést, valamint a quicksort két változatát – négyféle sorozaton: véletlenszerűn, majdnem rendezetten, visszafelé rendezetten, és csak néhány különböző értéket tartalmazón. A harminckét animáció egyszerre is lejátszható, de a csak adott rendezési algoritmushoz vagy adott típusú sorozathoz tartozók is külön megjeleníthetők.

A csak adott algoritmushoz tartozó, különböző típusú sorozatok rendezését bemutató animációk (3. ábra) segítségével alaposabban megérthető az algoritmus működése, valamint vizsgálható, hogy speciális sorozatok esetén eltérően teljesít-e a véletlenszerű sorozathoz képest. Egy rendezési algoritmus esetén szempont lehet például, hogy egy majdnem rendezett sorozattal hamarabb végezzen.

Insertion Sort



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

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



3. ábra A beillesztéssel rendezés hatékonyságának szemléltetése animáción különböző típusú sorozatok esetén [4]

Ha pedig az animációk egy adott típusú sorozat rendezését mutatják be különböző algoritmusok esetén, akkor eldönthető, hogy közülük melyeket érdemes vagy nem érdemes a gyakorlatban használni akkor, ha a rendezendő sorozatok várhatóan a vizsgált sorozat típusába fognak tartozni. Bizonyosfajta sorozatok esetén az úgynevezett hatékony algoritmusok meglepően rosszul teljesítenek. Végül kimondható, hogy nincsen legjobb rendezési algoritmus, vagyis amely minden típusú sorozaton a leggyorsabban végezne.

Habár a bemutatott honlapon futó program nem számolja meg az összehasonlításokat és az értékmásolásokat, könnyen készíthetők olyan programok, amelyek akár animáció nélkül, de meghatározzák ezeket az értékeket. Egy tízelemű véletlen sorozat rendezéséhez például az alábbiakban olvasható mennyiségű hasonlításra és másolásra volt szükség három rendezési algoritmus esetén.

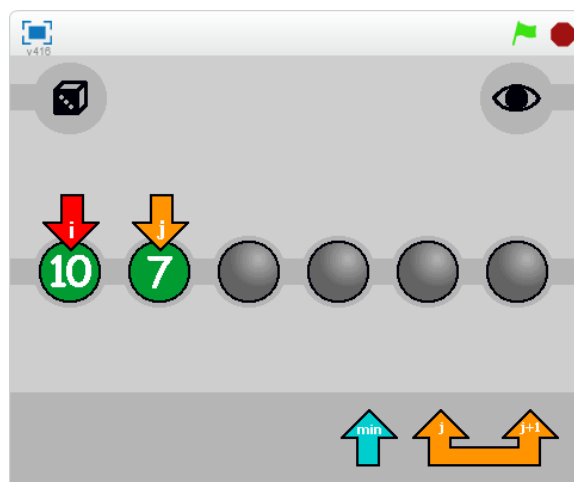
Rendezési algoritmus	Hasonlítások száma	Másolások száma	Összesen
Kiválasztásos rendezés	45	27	72
Beillesztéssel rendezés	35	46	81
Buborékredezés	45	84	129

Természetesen általában is meghatározható, hogy egy N elemű rendezendő sorozat esetén legalább és legfeljebb mennyi a hely- és műveletigénye a rendezési algoritmusoknak.

- ♦ Kiválasztásos rendezés
 - Helyfoglalás: $N+1$
 - Hasonlítások száma: $N*(N-1)/2$
 - Másolások száma: $3*(N-1)/2$
- ♦ Beillesztéses rendezés
 - Helyfoglalás: $N+1$
 - Hasonlítások száma: $N-1 - N*(N-1)/2$
 - Másolások száma: $O - 3*N*(N-1)/2$
- ♦ Buborékredezés
 - Helyfoglalás: $N+1$
 - Hasonlítások száma: $N*(N-1)/2$
 - Másolások száma: $O - 3*N*(N-1)/2$

3.3. Munkáltatás szimulációs programmal

A rendezési algoritmusokkal foglalkozó szimulációs programokban az algoritmusokat a felhasználóknak kell végrehajtaniuk. Habár ezek a programok a tárgyakkal kísérletezéshez képest kevesebb mozgás- és érzékszervvel vannak kapcsolatban, használatukhoz nincsen szükség tárgyi eszközökre és előkészületre, és akár mindenki egyénileg dolgozhat. Emellett pedig a tanári felügyeletet kiváltó vagy kiegészítő szolgáltatásokat is nyújthatnak: ellenőrizhetik az aktuálisan végrehajtott lépés helyességét, vagy folyamatos instruálással meg is taníthatják az algoritmust.



4. ábra Rendezés szimulációs programban

Egy általam készített programban (4. ábra) hatelemű sorozaton próbálhatók ki az alapalgoritmusok. Mozgatható, az algoritmizálás és a programozás során jellemzően használt i és j ciklusváltozókat szemléltető nyilakkal lehet rámutatni az elemekre azok megtekintéséhez vagy cseréjéhez. Ezek mellett a kiválasztásos rendezéshez felhasználható a min mutató, a buborékredezéshez pedig a j -edik és a $j+1$ -edik elemre is rámutató duplanyíl. A szem megjeleníti az összes elemet ellenőrizhetővé téve a megoldást, a kockára kattintáskor pedig újabb véletlen sorozatot kapunk.

Komolyabb alkalmazás érhető el az Algorithmics weboldalon [5], amely a rendezési módszereket néptáncosokkal bemutató videóiról híresült el. A beillesztéses

rendezés, a kiválasztásos rendezés, a buborékrendezés, a quicksort és a shell rendezés tekinthető meg és próbálható ki. Interaktív módban két elem kiválasztása után eldönthető, hogy azokat összehasonlítani vagy cserélni szeretnénk-e. A program figyelmeztet, ha az aktuális algoritmus szerint rossz lépést kívánunk végrehajtani. Egyetlen kattintással elrejtethők az elemértékek, hogy az összehasonlításra valóban szükség legyen. Ha pedig elakadnánk, a Help gomb megsúgja a következő lépést.

3.4. Tanítási célok teljesítése

A rendezési algoritmusok számítógépes megvalósítása valamilyen adatszerkezet célszerű használatát, egymásba ágyazott ciklusok szervezését, rekurzív eljárás készítését igényelheti. A rendezési algoritmusok a programozási tételek közül a legösszetettebbek: leírásuk és kódolásuk a gondolkodást próbára tévő feladatok, amelyek megoldását a képességektől függően lehet kisebb vagy nagyobb mértékben segíteni.

4. Összefoglalás

A rendezési algoritmusok megtanításához tehát számos módszer közül választhatunk. A tanulói sajátosságokon túl figyelembe vehetjük, hogy ezek a módszerek jellegüknél fogva a tanulás eltérő szakaszaiban lehetnek eredményesebbek. Az alsó tagozatban a különböző súlyú tárgyak az algoritmusok megértésében, begyakorlásában és egyszerű hatékonyságvizsgálatában is jól használhatók, a középiskolában azonban más módszerek javasolhatók az algoritmusok megértéséhez, leírásához és kódolásához, és hatékonyságvizsgálatához.

Az algoritmusok lényegének megértésében a képek bizonyulhatnak a leghasznosabbnak, mert csak a legfontosabb lépéseket emelik ki, a részletek megértésében viszont az összes lépést bemutató animációk. A megértés – a leírás és a kódolás előtt – a szimulációs programokkal ellenőrizhető a legjobban, amelyekkel be is gyakorolhatók az egyes algoritmusok. A leírást és a kódolást leginkább azok az animációk segítik, amelyek jelölik az algoritmusban jellemzően használt ciklusváltozókat és mutatókat, a hatékonyságvizsgálatot pedig értelemszerűen a hatékonyságot szemléltető animációk és programok.

Habár cikkemben a rendezési algoritmusok tanításában használható, vizualításra és cselekvésre épülő módszereket mutattam be, a teljes programozástanításban is törekednünk érdemes minél több hasonló módszer alkalmazására. Közben tartsuk szem előtt a kínai közmondást: „Hallom és elfelejtem. Látom és emlékszem rá. Csinálok és megértem.”

5. Irodalomjegyzék

- [1] I. Falus, Didaktika, Budapest: Nemzeti Tankönyvkiadó, 2006.
- [2] Sorting Algorithms, „Sorting Algorithms,” 2014. [Online]. Available: http://cow.ceng.metu.edu.tr/Courses/download_courseFile.php?id=5451. [Hozzáférés dátuma: 20. 05. 2014.].
- [3] Visualization of Data Structures, „Visualization of Data Structures,” 2014. [Online]. Available: <http://rosemarietan.com/fyp>. [Hozzáférés dátuma: 20. 05. 2014.].
- [4] Sorting Algorithms Animations, „Sorting Algorithms Animations,” 2014.. [Online]. Available: <http://www.sorting-algorithms.com/>. [Hozzáférés dátuma: 01. 05. 2014.].
- [5] Algorithemics, „Algorithemics,” 2014. [Online]. Available: <https://algorithemics.ms.sapientia.ro/>. [Hozzáférés dátuma: 20. 05. 2014.].