



Eötvös Loránd Tudományegyetem

Informatikai Kar

Algoritmusok és Alkalmazásaik Tanszék

Rendezési algoritmusok szemléltetése

Veszprémi Anna
mestertanár

Márföldi Péter Bence
programtervező informatikus BSc

Budapest, 2015

Tartalomjegyzék

1. Bevezető	1
1.1. A feladat és annak értelmezése	1
1.2. Definíciók és jelölések	1
1.3. Alkalmazott technológiák	1
1.3.1. A Java-ról röviden	1
1.3.2. JavaFX	2
2. Felhasználói dokumentáció	3
2.1. A vizsgált algoritmusok	3
2.1.1. Buborékrendeztés	3
2.1.2. Beszúró rendezés	3
2.1.3. Shell rendezés	3
3. Fejlesztői dokumentáció	4

1. fejezet

Bevezető

Az bizonyos, hogy minden informatikus - beleértve a leendőket is - tanulmányaik kezdetén találkoztak a rendezési algoritmusokkal. Nagyszerű terület arra, hogy megismerhessük a bonyolultság kérdését, azt hogy mi számít igazán sok adatnak, vagy éppen, hogy mikor nevezünk egy algoritmust stabilnak.

1.1. A feladat és annak értelmezése

1.2. Definíciók és jelölések

1.3. Alkalmazott technológiák

A Következőkben röviden összefoglaljuk a Java[1] és a JavaFX[2] jellegzetességeit.

1.3.1. A Java-ról röviden

A Java egy általános célú, objektumorientált programozási nyelv, melyet 2009-ig a *Sun Microsystems* fejlesztett, ezt követően pedig az *Oracle*. A szakdolgozatban használt 1.8-as verziót már az *Oracle* adta ki 2014-ben. A Java nyelv a szintaxisát a C és C++ nyelvektől örökölte, azonban utóbbitól eltérően egyszerű objektummodellel rendelkezik.

A Java platformra készült programok túlnyomó többsége asztali alkalmazás. Ma-napság egyre több helyen találkozhatunk a Java nyelven írt programokkal, például mobil eszközökön, banki rendszereknél vagy akár egy szórakoztató elektronikai eszközön. Nagy előnye, hogy sok nyelvvel ellentétben platformfüggetlen, azaz egy adott platformról egy program minimális változtatással átültethető egy másik platformra.

A Java legfontosabb része a *Java virtuális gép (JVM)*. A *JVM*-et sokféle be-
rendezés és szoftvercsomag tartalmazza, így a nyelv egyaránt platformként és kö-

zép szintként is működik. Összefoglalva a Java program három fontos szerepet tölt be:

- **programozási nyelv**
- **köztes réteg (middleware)**
- **platform**

1.3.2. JavaFX

Olyan szoftverplatform, amelynek célja, hogy gazdag internetes alkalmazást lehessen készíteni és futtatni eszközök széles skáláján. Eredetileg a *Swing* könyvtárat váltotta volna fel, azonban jelenleg mindkettő része a *Jave SE*-nek.

A 2.0-ás verzióig a fejlesztők egy külön nyelvet használtak, amelyet *JavaFX Script*-nek neveznek. Azonban mivel ez szintén Java bájt-kódot generál a későbbiekben megadott a lehetőség, hogy a programozók Java kódot használjanak helyette. A JavaFX egyik legnagyobb előnye, hogy egy egyszerű *XML* struktúrában leírhatók a program grafikus felületének összetevői, melyhez ezt követően elegendő az egyes interakciókhoz tartozó funkciókat implementálni.

Az elterjedtebb operációs rendszerek mindegyikét támogatja. Ahogyan előnye, úgy hátránya is a *Swing*-hez képest az, hogy jelenleg is folyik a fejlesztése, ezért olykor csak hosszas utánajárást követően sikerül megoldást találni egy-egy problémára.

2. fejezet

Felhasználói dokumentáció

2.1. A vizsgált algoritmusok

2.1.1. Buborékredezés

A legrégebbi és a legegyszerűbb rendezési algoritmus. Mindemellett a legtöbb esetben a leglassabb is. Már az 1965-ös évben megjelent egy teljes körű elemzése[3].

A rendezés minden egyes elemet összehasonlít a rákövetkező elemmel, és ha szükséges megcseréli őket. Egészen addig, amíg nincs egy olyan menet, amelyben egyetlen elem sem cserél helyet. Ez azt eredményezi, hogy lépésenként a maximális elem "buborék" szerűen a lista végére kerül, ezzel egyidejűleg a kisebb elemek "lesüllyednek" a tömb elejére. Az algoritmus javítható azzal, hogy nem vizsgáljuk meg mindig a tömb összes elemét, hanem amennyiben egy maximális elem elérte a helyét visszavezetjük a problémát az eggyel "rövidebb" rendezési feladatra[4].

2.1.2. Beszűrő rendezés

A nevéből egyszerűen kikövetkeztethető az eljárás - beszűrja az elemeket a megfelelő helyükre a végleges tömbbe. A legegyszerűbb megvalósítása két tömböt használ, a "forrás" tömböt és egyet, amelyben az elemek már rendezve vannak. Memóriamegtakarítás céljából a leggyakrabban a helyben rendező változatát valósítják meg. Ennek lényege, hogy a soron következő elemet egy ideiglenes változóba elmentünk, és a rendezett tömb elemeit jobbra csúsztatjuk, mindaddig amíg a kiválasztott érték nem kerül a helyére. Kezdetben a tömb első elemét tekintjük rendezettnek. A beszűrő rendezés sok esetben a leggyorsabb négyzetes idejű rendezés.

2.1.3. Shell rendezés

3. fejezet

Fejlesztői dokumentáció

Irodalomjegyzék

- [1] *Java (programming language)*, Wikipedia the free encyclopedia. [ONLINE] [Hivatkozva: 2015.04.21] [http://en.wikipedia.org/wiki/Java_\(programming_language\)/](http://en.wikipedia.org/wiki/Java_(programming_language)/)
- [2] *JavaFX*, Wikipedia the free encyclopedia. [ONLINE] [Hivatkozva: 2015.04.21] <http://en.wikipedia.org/wiki/JavaFX/>
- [3] Demuth, H.: *Electronic Data Sorting*, PhD thesis, Stanford University, 1956, [184]
- [4] Dr. Fekete István.: *Algoritmusok és adatszerkezetek I. jegyzet*, [ONLINE] [Hivatkozva: 2015.04.20] http://people.inf.elte.hu/fekete/algoritmusok_bsc/alg_1_jegyzet/