



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
2. Felhasználói dokumentáció	2
2.1. A vizsgált algoritmusok	2
2.1.1. Buborékrendeztés	2
2.1.2. Beszúrórendezés	2
3. Fejlesztői dokumentáció	3

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

1.3.1. A Java-ról röviden

A Java egy általános célú, objektumorientált programozási nyelv. 2009-ig a *Sun Microsystems* fejlesztette, 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. Manapság 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.

2. fejezet

Felhasználói dokumentáció

2.1. A vizsgált algoritmusok

2.1.1. Buborékrendezés

A legrégebbi és a legegyszerűbb rendezési algoritmus. Mindemellett a legtöbb esetben a leglassabb is. Már az 1965-ös évben megjelent egy teljes körű elemzése[1].

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[2].

2.1.2. Beszűrőrendezés

3. fejezet

Fejlesztői dokumentáció

Irodalomjegyzék

- [1] Demuth, H.: *Electronic Data Sorting.*, PhD thesis, Stanford University, 1956, [184]
- [2] Dr. Fekete István.: *Algoritmusok és adatszerkezetek I. jegyzet*, [ONLINE] [Hivatkozva: 2015.04.20] http://people.inf.elte.hu/fekete/algoritmusok_bsc/alg_1_jegyzet/