Ohjelman yleisrakenne

Ohjelmassa on luokat taulukon luomista ja järjestämistä varten. ArrayGenerator-luokalla voi luoda haluamansa kokoisen taulukon täytettynä luvuilla. Luvut voivat olla joko järjestyksessä ykkösestä suurimpaan tai käänteisessä järjestyksessä suurimmasta luvusta ykköseen. Lisäksi vaihtoehtona on luoda taulukko satunnaisista luvuista, jolloin käyttäjä voi määrätä mikä on suurin luku, joka taulukkoon tulee. Sorter-luokan avulla tällainen taulukko voidaan järjestää käyttäjän valitsemalla algoritmilla. Valittavana on pikajärjestäminen (quicksort), lomitusjärjestäminen (merge sort),voit kekojärjestäminen (heap sort), lisäysjärjestäminen (insertion sort), kuplajärjestäminen (bubble sort) ja bogo sort.

Aikavaativuudet ja tilavaativuudet

Pikajärjestäminen

Saavutetut aikavaativuudet ovat alkuperäisten tavoitteiden mukaiset: parhaimmassa tapauksessa O(n log n) ja huonoimmassa $O(n^2)$. Tässä toteutuksessa huonoin kuitenkin toteutuu useammin, koska jakoalkioksi valitaan aina taulukon viimeinen luku eikä esimerkiksi kolmen luvun mediaania. Tilavaativuus on O(n), joka olisi voinut olla parempi O(n log n)

Lomitusjärjestäminen

Koska merge-metodin aikavaativuus on O(n) ja taulukko jaetaan aina kahteen osaan, jolloin kokonaisaikavaativuus on O(n log n). Tilavaativuus O(n).

Kekojärjestäminen

Koska heapify-metodin aikavaativuus O(log n). Tällöin itse järjestäminen tapahtuu ajassa O(n log n), jolloin tavoiteltu aikavaativuus on saavutettu. Sama aikavaativuus pätee kaikkeen tilanteisiin. Tilavaativuus O(1).

Lisäysärjestäminen

Pahimman tapauksen aikavaativuus $O(n^2)$ toteutuu kahden sisäkkäisen silmukan takia. Parhaimmassa tapauksessa algoritmi vain vertaa kahta alkiota toisiinsa eikä niitä tarvitse siirtää, jolloin aikavaativuus on O(n). Tilavaativuus O(1).

Kuplajärjestäminen

Aikavaativuus on kaikissa tapauksissa $O(n^2)$ kahden sisäkkäisen silmukan takia. Tilavaativuus O(1).

Bogosort

Pahinta tapausta ei voi määrittää algoritmin satunnaisuuden vuoksi. Parhaimmassa tapauksessa aikavaativuus on O(n). Keskimääräinen aikavaativuus on O(n*n!), koska keskimäärin järjestyksiä käydään läpi n! ja jokaisen järjestyksen tarkistamiseen aikaa kuluu O(n).

Lähteet

www.cs.helsinki.fi/u/floreen/tira2015/sivut351-638.pdf
T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein: *Introduction to Algorithms*
http://en.wikipedia.org/wiki/Bogosort