

Tiralabrassa tulen ratkaisemaan taulukonjärjestämisen eri järjestämisalgoritmien avulla:

- Kekojärjestäminen
 - Aikavaativuus kekojärjestämiselle on $O(n \log n)$, ja kekojärjestämisessä käytetyn heapify-operaation rekursioiden takio tilavaativuus on $O(\log n)$.
- Kupla järjestäminen
 - Aikavaativuus on $O(n^2)$ ja tilavaativuus tälläkin $O(1)$ koska aputilaa ei tarvita
- Lomitusjärjestäminen
 - Aikavaativuus on $O(n \log n)$ ja tilavaativuus on $O(n)$
- Pikajärjestäminen
 - Aikavaativuus on pahimmassa tapauksessa $O(n^2)$, keskimääräisessä tapauksessa kuitenkin $O(n \log n)$ ja tilavaativuus on $O(\log n)$
- Laskemisjärjestäminen
 - Aika vaatavuus on $O(n + k)$, missä n = alkioden määrä ja k = mahdollisten numeroiden määrä, tilavaativuus on sama $O(n + k)$

Tulen kaikissa järjestämisalgoritmeissa pyrkimään yllä mainittuihin aika- ja tilavaativuuksiin.

Lopulta vertailen eri algoritmien nopeutta toisiaan ja Javan valmista `Arrays.sort()` metodia vastaan ajamalla ne läpi n kertaa ja poimimalla joka kerta järjestämisen keston millisekunneissa ja lopulta katsotaan kaikkien keskimääräinen suorittamisaika.

Tietorakentaana tulen tekemään maksimikekoan, sekä minimikeon.

Lähteet:

Lisäilen tänne lähteitä sitä mukaan kun niitä tulee

Tiran materiaalit

StackOverflow sivusto, arrayn max kokoa ihmetellessä