

## 58131 Tietorakenteet ja algoritmit (kevät 2015)

### Harjoitus 3 (26.–30.1.2015)

1. Annettuna on  $n$  luvun taulukko, joka on järjestyksessä pienimmästä suurimpaan. Tehtävänä on selvittää ajassa  $O(\log n)$ , kuinka monta kertaa luku  $x$  esiintyy taulukossa. Esim. jos taulukko on  $[2, 3, 3, 4, 4, 4, 5]$  ja  $x$  on 4, oikea vastaus on 3.  
Suunnittele algoritmi tehtävään ja toteuta se Javalla. Testaa, että ohjelmasi toimii oikein.
2. Ohjelmoi jokin yksinkertainen järjestämisalgoritmi. Kuinka suuren taulukon sillä pystyy järjestämään sekunnissa?  
Testaa sitten Javan metodia `Arrays.sort` (tai muun kielen vastaavaa). Kuinka suuren taulukon pystyt järjestämään sillä sekunnissa?
3. Järjestämme  $n$  lukua sisältävän kokonaislukutaulukon nousevaan järjestykseen lisäysjärjestämisellä. Mikä on algoritmin aikavaativuus seuraavissa tapauksissa? Perustele.
  - (a) Taulukossa on  $n$  erisuurta lukua ja ne ovat alun perin nousevassa suuruusjärjestyksessä.
  - (b) Taulukossa on  $n$  erisuurta lukua ja ne ovat alun perin alenevassa suuruusjärjestyksessä.
  - (c) Taulukossa on yksi ja sama luku  $n$  kertaa.
4. Mitkä ovat vastaukset (a), (b) ja (c), jos vaihdamme edellisessä tehtävässä järjestämisalgoritmin lomitusjärjestämiseksi?
5. Mitkä ovat vastaukset (a), (b) ja (c), jos vaihdamme edellisessä tehtävässä järjestämisalgoritmin pikajärjestämiseksi luennolla esitettyä partition-funktiota käyttäen (jakoalkio on ensimmäinen luku)?
6. Taulukon järjestäminen onnistuu ajassa  $\mathcal{O}(n \log n)$ , kun taulukossa on  $n$  alkia. Entä kuinka nopeasti taulukon voi sekoittaa? Anna tehokas algoritmi taulukon sekoittamiseen. Tee algoritmisi ajankäytöstä optimaalinen. Mikä on algoritmisi aikavaativuus ja tilavaativuus? Voit olettaa, että käytössäsi on funktio `random(i, j)`, joka palauttaa satunnaisen kokonaisluvun väliltä  $[i, j]$  vakioajassa. Sekoittaako algoritmisi taulukon niin, että kaikki lukujen järjestykset ovat yhtä todennäköisiä?