

58131 Tietorakenteet ja algoritmit (kevät 2015)

Harjoitus 4 (2.–6.2.2015)

1. Sinulla on n kappaletta erikokoisia pultteja ja n kappaletta niihin sopivia mutte-reita. Saat kokeilla, sopiiko mutteri pulttiin, mistä voit päätellä, onko mutteri liian suuri, liian pieni vai juuri sopivan kokoinen. Et voi kuitenkaan verrata kahta pulttia tai kahta mutteria keskenään. Tehtävänäsi on löytää jokaiselle pultille sopiva mutteri. Suunnittele algoritmi, joka toimii odotusarvoisesti ajassa $\mathcal{O}(n \log n)$. Tarkkaa pseudokoodia tai todistusta aikavaativuudelle ei vaadita.
2. Pinon ja jonon voi toteuttaa Javassa esim. tyyppien `ArrayDeque` ja `LinkedList` avulla. Ota selvää, mihin luokkien toteutus perustuu. Tee vielä jokin testi siitä, kuinka tehokkaita luokat ovat käytännössä. (Voit esim. lisätä jonoon n lukua ja sitten poistaa ne yksitellen.)
3. Java tarjoaa esim. luokat `ArrayList` ja `LinkedList` sen tyyppisiin tehtäviin, joihin luennolla esitetty abstrakti tietotyyppi *joukko* on tarkoitettu. Mille luennolla esite-tyistä operaatioista nämä luokat tarjoavat (jokseenkin) valmiin tehokkaan toteutuk-sen? Mikä on toteutuksen perusidea? Mitkä ovat operaatioiden aikavaativuudet? Tee joitain yksinkertaisia kokeita ja tarkastele suoritusajoja erilaisilla jonoilla operaa-tioita lisäys, haku ja poisto. Löydätkö tilanteita, joissa `ArrayList` on tehokkaampi kuin `LinkedList` tai toisin päin?
4. Syötteenä annetaan kaksi yhteen suuntaan linkitettyä tunnussolmullista järjestettyä listaa M ja N . Kussakin listassa alkiot ovat erillisiä kokonaislukuja (siis sama luku ei esiinny monta kertaa listassa M , mutta listassa M oleva luku voi olla myös listassa N). Anna algoritmi, joka tuottaa yhdistetyn listan L , joka on sekin yhteen suuntaan linkitetty tunnussolmullinen järjestetty lista.
Yhdistetyssä listassa kukin luku esiintyy vain kerran. Älä oleta mitään listojen pi-tuudesta (M tai N voi esimerkiksi olla tyhjä). Yhdistämisen jälkeen ei tarvitse enää ylläpitää syötteenä annettuja listoja M ja N .
Algoritmisi tulee toimia vakiotilassa. Käytä osoittimia. Et voi olettaa, että listallasi on mitään valmiita operaatioita (`insert`, `delete`, ...).
Esimerkki: jos M sisältää luvut 1, 3, 5, 6, 12, 14 ja N sisältää luvut 5, 7, 12 niin tuloksena on lista L , joka sisältää luvut 1, 3, 5, 6, 7, 12, 14.
5. Näytä, miten jonoa on mahdollista simuloida kahden pinon avulla: Toteuta ope-raatiot `enqueue(Q, k)`, `dequeue(Q)` ja `empty(Q)`. Oleta jono ja pinot rajattoman kokoisiksi. Mikä on simuloidun jonon kunkin operaation aikavaativuus?
6. Näytä, miten pinoa on mahdollista simuloida kahden jonon avulla: Toteuta operaatiot `push(S, k)`, `pop(S)` ja `empty(S)`. Oleta pinot ja jono rajattoman kokoisiksi. Mikä on simuloidun pinon kunkin operaation aikavaativuus?
7. *Vapaaehtoinen lisätehtävä* Voiko yhdellä pinolla simuloida yhtä jonoa käyttäen sen lisäksi vain vakiomäärän muistia? Entä voiko yhdellä jonolla simuloida yhtä pinoa? Aikavaativuuksia ei tarvitse ottaa huomioon. Tarkkaa pseudokoodia ei tarvitse esit-tää.