

Tietorakenteet, laskuharjoitus 10, 4-8.4

1. Seuraavissa ongelmissa algoritmin syötteenä on taulukko T , jossa on joukko kokonaislukuja. Esitä kuhunkin ongelmaan ajassa $O(n \log n)$ toimiva ratkaisualgoritmi.

- (a) Esiintyykö jokin luku taulukossa monta kertaa?
- (b) Mikä luku esiintyy taulukossa useimmiten?
- (c) Mikä on pienin ero kahden taulukossa olevan luvun välillä?

2. Algoritmi saa syötteen taulukon T , jossa on joukko kokonaislukuja sekä kokonaisluvun k . Esitä ajassa $O(n \log n)$ toimiva algoritmi, joka tutkii onko taulukossa kahta lukua, joiden summa on k ? Oletetaan tässä, että sama taulukon luku ei saa kuulua summaan kahta kertaa.

Lisähaaste (ei edellytetä rastiin):

- (a) Esitä ajassa $O(n^2)$ toimiva algoritmi, joka tarkistaa, onko taulukossa kolmea lukua, joiden summa on k .
- (b) Esitä ajassa $O(n^2 \log(n))$ toimiva algoritmi, joka tarkistaa, onko taulukossa neljää lukua, joiden summa on k .

Oletetaan myös näissä, että sama taulukon luku ei saa kuulua summaan kuin kerran.

3. Toteuta haluamallasi ohjelmointikielellä lomituserjästäminen tai pikajärjestäminen tai mieluiten molemmat.

Java API:n `Arrays`-luokassa on määritelty metodi `sort` jonka avulla on mahdollista järjestää taulukollinen mitä tahansa olioita¹.

Vertaile tällä ja edellisellä viikolla toteuttamiesi järjestämisalgoritmien sekä Javan `Arrays.sort`:in suorituskykyä erikokoisilla suurilla syötteillä, tapauksissa, joissa syötteenä oleva taulukko on

- satunnaisessa järjestyksessä
- valmiina järjestyksessä
- valmiina käänteisessä järjestyksessä

Jos käytät jotain muuta kieltä kuin Javaa, ota vertailukohdaksi kielesti standardissa oleva järjestämisalgoritmiteotus.

Huom: jos toteutat pikajärjestämisen suoraan sivujen 360 ja 362 mukaan, pahin tapaus sisältää hyvin syvän rekursion ja joudut kasvattamaan ajoaikaisen pinon kokoa välttyäksesi `java.lang.StackOverflowError`:ilta. Pinoa voi kasvattaa antamalla käynnistyksen yhteydessä parametrin `-Xss`, jonka yhteydessä määritellään ajonaikaisen pinon koko. Esim. `java -Xss100m Quicksort` määrittelee pinon kooksi 100 megatavua.

Jos valitset partition-operaation jakoalkioksi a keskimmäiseksi suurimman arvoista $A[p]$, $A[(p+q)/2]$ ja $A[q]$ välttyt pahimmalta tapaukselta jos taulukko on valmiiksi järjestetyssä.

Viime viikon tapaan tee algoritmistasi sellaisia että ne toteuttavat rajapinnan

```
public interface Sort {
    void sort(int[] table);
}
```

¹olioiden on toteutettava rajapinta `Comparable` tai niille on määriteltävä `Comparator`-rajapinnan toteuttava vertailija. Alkeistyyppien käärinluokat `Integer`, `Long` ja `Double` sekä `String` toteuttavat `Comparable`-rajapinnan.

Tällöin voit käyttää suorituskykyvertailuissa valmista luokkaa osoitteesta:

<https://wiki.helsinki.fi/display/javak11/Jarjestysalgoritmien+suorituskykymittaja>

4. Taulukon järjestäminen onnistuu ajassa $O(n \log n)$. Entä kuinka nopeasti taulukon voi sekoittaa? Määrittele tehokas algoritmi taulukon sekoittamiseen. Mikä on algoritmisi aikavaativuus ja tilavaativuus? Voit olettaa, että käytössäsi on funktio, joka palauttaa satunnaisen kokonaisluvun väliltä $1 \dots n$ vakioajassa. Sekoittaako algoritmisi taulukon niin, että kaikki lukujen järjestykset ovat yhtä todennäköisiä?
5. Suunnatun verkon $G = (V, E)$ transpoosi G^T saadaan vaihtamalla kaikkien verkon G kaarten suunta. Siis $G^T = (V, E^T)$, missä $E^T = \{(v, u) \mid (u, v) \in E\}$. Esitä ajassa $O(|V| + |E|)$ toimiva algoritmi, joka muodostaa verkon G vieruslistaesityksen perusteella vieruslistaesityksen transpoosille G^T .