

Tehtävä 2: Lajittelualgoritmit

Suttupaperi

Voit käyttää alla olevaa tekstilaatikkoo suttupaperina. Saat irrotettua suttupaperin “kelluvaksi” laatikoksi tekstilaatikon oikeassa yläreunassa sijaitsevasta nuolipainikkeesta. Voit siirrellä kelluvaa ikkunaa yläpalkista ja muuttaa sen kokoa oikeasta alanurkasta. Voit sitten selata tehtävän kysymysten kohdalle ilman, että suttupaperi katoaa näkyvistä. Suttupaperin sulkeminen palauttaa sen takaisin tämän ohjeen alle.

Avaa suttupaperi ja laskin toisessa välilehdessä

Tätä voit käyttää suttupaperina

Aineisto

Lajittelualgoritmien perusteet

Lajittelualgoritmit ovat tietokoneohjelmia, jotka järjestävät joukon tietoja, esimerkiksi numeroita tai kirjaimia, tietyllä tavalla nousevaan tai laskevaan järjestykseen. Lajittelualgoritmeja käytetään laajasti tietojenkäsittelyssä, tilastotieteessä ja tietokoneohjelmoinnissa. Yleisesti tunnettuja lajittelualgoritmeja ovat esimerkiksi lomitustajittelu, pikatajittelu, kuplatajittelu ja vaihtolajittelu. Jokaisella algoritmilla on omat vahvuutensa ja heikkoutensa, joten valinta riippuu käytettävän datan määrästä ja sen luonteesta sekä käytettävissä olevista resursseista.

Lajittelualgoritmien nopeudessa voi olla huomattavia eroja etenkin isoilla datamäärillä. **Kuplatajittelu** ja **vaihtolajittelu** eivät ole yhtä tehokkaita kuin monet muut lajittelualgoritmit, kuten pikatajittelu ja lomitustajittelu, mutta ne ovat helpommin ymmärrettävissä ja voivat olla käyttökelpoisia, jos lajiteltavien alkioiden määrä on kohtuullisen pieni.

Lajittelualgoritmeja käytettäessä lajiteltavat tiedot tallennetaan usein **taulukko**on. Taulukot ovat ohjelmoinnissa käytettyjä tietorakenteita, jotka mahdollistavat useiden saman tyyppisten arvojen tallentamisen. Taulukoita käytetään usein silloin, kun on tarve käsitellä suurta määrää tietoa, joka on järjestetty tietynlaiseen loogiseen ryhmään.

Taulukot koostuvat alkioista, joilla jokaisella on oma sijaintinsa taulukossa. Alkioiden sijainti määritellään numerolla, jota kutsutaan indeksiksi. Taulukon ensimmäinen alkio on indeksissä 0, toinen indeksissä 1 ja niin edelleen.

Indeksi	0	1	2	3	jne
Alkio	12	4	-34	34	...

Ohjelmointikielissä taulukon indekseihin viitataan yleisesti hakasulkeiden sisään sijoitetulla numerolla. Yllä olevan taulukon ensimmäisessä indeksissä 0 on alkio, jonka arvo on 12. Toisessa indeksissä 1 on alkio jonka arvo on 4, ja niin edelleen.

Seuraavassa eräällä ohjelmointikielellä laaditussa esimerkissä muuttujaan `eka_luku` sijoitetaan taulukon ensimmäisen alkion arvo. Tämän jälkeen muuttujan arvo tulostetaan näytölle `print()`-komennolla. Voit kokeilla taulukon eri alkioiden arvojen tulostamista muuttamalla indeksia hakasulkeiden sisällä rivillä 2.

Huomaathan, että jos viittaat ohjelmassa indeksiin, jota ei ole taulukossa (esimerkiksi alla olevassa taulukossa 4 tai -1), ohjelma tuottaa virheilmoituksen `undefined`. Tämä johtuu siitä, että viittaus tapahtuu taulukon indeksien arvoalueen ulkopuolelle, toisin sanoen indeksiin, jota ei ole taulukossa olemassa.

Esimerkki taulukon yksittäiseen alkioon viittaamisesta

Voit kokeilla esimerkkiä muuttamalla haettavan alkion indeksia rivillä 2.

```
1 taulukko = [12, 4, -34, 34]
2 eka_luku = taulukko[0]
3 print(eka_luku) // tulostaa luvun 12, jos rivillä 2 on indeksi 0
```

Kupla- ja vaihtolajittelut käyttävät syötteenään yllä kuvattua taulukkomuotoista tietorakennetta. Taulukon alkioita käydään läpi algoritmin määrittelemällä tavalla, ja lopputuloksena on nousevaan järjestykseen järjestetty taulukko.

Kuplalajittelu

1. Algoritmi aloittaa (kierros 1) vertaamalla ensimmäistä (indeksissä 0) ja toista (indeksissä 1) alkioita taulukossa. Jos ne ovat väärässä järjestyksessä, eli taulukon ensimmäisenä oleva alkio on isompi kuin verrattava alkio, niiden paikkoja vaihdetaan.
2. Algoritmi siirtyy taulukon seuraavaan lukupariin (toinen, eli indeksi 1 ja kolmas alkio taulukossa, eli indeksi 2) ja vertaa niitä. Jos ne ovat väärässä järjestyksessä, niiden paikkoja vaihdetaan. Näin edeten (kolmas ja neljäs alkio, neljäs ja viides alkio jne.) käydään koko taulukko läpi.
3. Tämän jälkeen algoritmi aloittaa vertailun uudelleen taulukon alusta (kierros 2,... jne) (eli vertaillaan ensimmäinen ja toinen alkio), ja käy taulukon uudelleen läpi. Taulukon läpikäyntiä (eli kierroksia) toistetaan niin kauan, kunnes kierroksen aikana ei tule enää yhtään vaihtoa.

Esimerkki kuplalajittelun etenemisestä. Jokaisella rivillä esitetään taulukon sisältö peräkkäisten alkioiden vaihdon jälkeen. Vaihto tehdään vain jos alkiot ovat väärässä järjestyksessä.

kierros	0	1	2	selitys
	18	13	11	Alkutila
1	18	13	11	Vaihdetaan alkiot indekseissä 0 ja 1
1	13	18	11	Vaihdetaan alkiot indekseissä 1 ja 2

kierros	0	1	2	selitys
2	13	11	18	Vaihdetaan alkiot indekseissä 0 ja 1
3	11	13	18	Lajiteltu taulukko, vaihtoja ei tarvitse enää tehdä

Vaihtoja tehtiin 3 kappaletta.

Kysymyksessä 2.2, jossa kysytään miten vastaava lajittelu toimisi, niin vastaus kirjoitetaan tässä tapauksessa alla olevassa muodossa. Huomaa että alkutilaa ei toisteta kahta kertaa kuten yllä olevassa taulukossa.

```
18 13 11
13 18 11
13 11 18
11 13 18
```

Vaihtolajittelu

1. Algoritmi aloittaa vertaamalla taulukon ensimmäistä alkioita (indeksissä 0) vuorollaan kaikkiin taulukon myöhempisiin alkioihin (indekseissä $1..n-1$, missä n = taulukon alkioiden lukumäärä). Jos verrattavat alkiot ovat väärässä järjestyksessä (taulukon sillä hetkellä ensimmäisenä oleva alkio on isompi kuin verrattava alkio), vaihdetaan näiden paikkaa. Kun taulukko on käyty kertaalleen läpi, taulukon ensimmäisenä on nyt kaikkein pienin alkio.
2. Algoritmi siirtyy vertaamaan taulukon toista (indeksissä 1) alkioita myöhempisiin alkioihin (indekseissä $2..n-1$), ja alkioiden paikkaa vaihdetaan jos ne ovat väärässä järjestyksessä.
3. Näin edetään, kunnes verrataan taulukon toiseksi viimeistä ja viimeistä alkioita. Kun nämä alkiot on verrattu ja tarvittaessa vaihdettu keskenään, on taulukko järjestyksessä.

Esimerkki vaihtolajittelun etenemisestä. Jokaisella rivillä esitetään taulukon sisältö alkioiden vaihdon jälkeen.

kierros	0	1	2	selitys
	18	13	11	Alkutila
1	18	13	11	Vaihdetaan alkiot indekseissä 0 ja 1
1	13	18	11	Vaihdetaan alkiot indekseissä 0 ja 2
2	11	18	13	Vaihdetaan alkiot indekseissä 1 ja 2
3	11	13	18	Lajiteltu taulukko, koska verrattu toiseksi viimeinen ja viimeinen

Vaihtoja tehtiin 3 kappaletta.

Vaihto-operaatio

Lajittelualgoritmien toiminnassa on keskeistä kahden listan alkion vaihtaminen keskenään. Yllä olevassa esimerkissä 2. rivillä taulukon kaksi ensimmäistä alkioita (18 ja 13) vaihdetaan keskenään. Käytännössä tämä tulee toteuttaa ohjelmointikielissä siten, että toinen vaihdettavista alkioista tallennetaan väliaikaiseen muuttujaan. Alla on esitys vaihto-operaatiosta erään ohjelmointikielen koodina. Esimerkissä taulukon nimenä on `lista`.

```
temp = lista[0]
lista[0] = lista[1]
lista[1] = temp
```

Koska sama tehtävä toistuu lajittelussa useita kertoja, tehdään ohjelmoinnissa usein niin, että toistuvaa tehtävää varten kirjoitetaan oma aliohjelma (funktio). Tehdään se tässä tapauksessa nimelle **swap**. Aliohjelmalle annetaan kolme parametria suluissa:

- käsiteltävän taulukon nimi
- ensimmäisen vaihdettavan alkion indeksi
- toisen vaihdettavan alkion indeksi

Silloin edellisen esimerkin vaihto saataisiin tehtyä kutsulla:

```
swap(lista, 0, 1)
```

Huomaa että **swap** vaihtaa alkiot keskenään riippumatta siitä ovatko ne järjyksessä vaiko ei.

Esimerkki aliohjelmasta swap

Vaihdetaan **lista**-nimisen taulukon paikoissa (indekseissä) 0 ja 1 olevat alkiot keskenään. Tulostetaan lista ennen ja jälkeen vaihtamisen.

```
1 lista = [13, 47, 1]
2 print(lista)
3 swap(lista, 0, 1)
4 print(lista)
```

Kysymykset

Kysymys 2.1 (0-8 pistettä)

Kuinka monta kertaa yllä kuvatut lajittelualgoritmit suorittavat alkioden vaihto-operaation mainitulle taulukolle?

a) 12, 6, 27, -4

Kuplajajittelu

Vaihtolajittelu

b) -23, 143, 13, 3, 32

Kuplajajittelu

Vaihtolajittelu

c) 143, 45, 32, 11, -10

Kuplalajittelu

Vaihtolajittelu

d) 1, 3, 5, 7, 9, 11, 13, 15, 19, 21

Kuplalajittelu

Vaihtolajittelu

Kysymys 2.2 (0-10 pistettä)

Kysymys 2.2.1 – Kuplalajittelu (0-4 pistettä)

Lajittele alla oleva kokonaislukutaulukko **kuplalajittelua** käyttäen nousevaan järjestykseen. Kirjoita vastauksesi vaihe kerrallaan materiaalin esimerkin mukaisesti tekstikenttään siten, että yhdellä rivillä on taulukon tila kunkin **vaihdoksen** jälkeen. Kirjoita toiselle riville taulukon tila ensimmäisen vaihdon jälkeen, kolmannelle riville toisen vaihdon jälkeen jne.

Ensimmäisellä rivillä on valmiiksi taulukon alkutila: 42 52 87 23

Muista jättää myös tämä alkutila omaan vastaukseesi.

1 42 52 87 23

Kysymys 2.2.2 – Vaihtolajittelu (0-6 pistettä)

Lajittele alla oleva kokonaislukutaulukko **vaihtolajittelua** käyttäen nousevaan järjestykseen. Kirjoita vastauksesi vaihe kerrallaan tekstikenttään siten, että yhdellä rivillä on taulukon tila kunkin **vaihdoksen** jälkeen. Kirjoita siis toiselle riville taulukon tila ensimmäisen vaihdon jälkeen, kolmannelle riville toisen vaihdon jälkeen jne.

Ensimmäisellä rivillä on valmiiksi taulukon alkutila: 154 143 85 45 32 13 8

Muista jättää myös tämä alkutila omaan vastaukseesi.

1 154 143 85 45 32 13 8

Kysymys 2.3 (0-12 pistettä)

Laadi alla olevalla koodieditorilla ohjelma, joka toteuttaa ensimmäisellä rivillä valmiina olevan listan lajittelun mainitulla lajittelualgoritmillä nousevaan järjestykseen. Voit lisätä komentoja koodiin editorin alla olevilla painikkeilla. `swap()`-komento vaihtaa listassa parametrien osoittamien indeksien alkiot keskenään. Saat käyttää tehtävän ratkaisussa vain painikkeissa määritettyjä käskyjä. Mikäli haluat poistaa lisäämiäsi komentoja, pyyhi rivin sisältö ihan normaalin tekstinkäsittelyn tavoin. **Komentoja saa olla vain yksi jokaisella rivillä.**

Esimerkki:

```
lista = [1, 9, 7]
swap(lista, 0, 2)    // nyt lista on [7, 9, 1]
```

Jos halut tehdä itsellesi “muistiinpanoja” voit tehdä sen `//`-jonolla kuten em. esimerkissä.

Komennolla `print(lista)` voit tulostaa listan näkyville. `print`-rivien käyttö ei vaikuta tehtävän pisteytykseen.

Kysymys 2.3.1 - Kuplalajittelu (0-4 pistettä)

Laadi ohjelma siten, että lista järjestetään **kuplalajittelulla** nousevaan järjestykseen.

`print(lista)`-komennolla voit tulostaa listan sen hetkisen sisällön.

Listan alkutila: 13, 47, 1

```
1 lista = [13, 47, 1]
```

Kysymys 2.3.2 - Vaihtolajittelu (0-8 pistettä)

Laadi ohjelma siten, että lista järjestetään **vaihtolajittelulla** nousevaan järjestykseen.

Tässä kysymyksessä **et saa käyttää** `print()`-komentoa.

```
1 lista = [12, -2, 1, 10, 8]
```