

Tietorakenteet 2018  
Harjoitukset 1, ratkaisut (Viikko 37)

---

**Algorithm 1** Maximum( $A[0, 1, \dots, n - 1]$ )

---

1. **alkuehto:**  $n \geq 1$ 
    - 1:  $a \leftarrow A[0]$
    - 2: **for**  $i \leftarrow 1$  **to**  $n - 1$  **do**
    - 3:   **if**  $a \leq A[i]$  **then**
    - 4:      $a \leftarrow A[i]$
    - 5:   **end if**
    - 6: **end for**
    - 7: **return**  $a$
- 

---

**Algorithm 2** Minimum( $A[0, 1, \dots, n - 1]$ )

---

- alkuehto:**  $n \geq 1$
- 1:  $a \leftarrow A[0]$
  - 2: **for**  $i \leftarrow 1$  **to**  $n - 1$  **do**
  - 3:   **if**  $A[i] \leq a$  **then**
  - 4:      $a \leftarrow A[i]$
  - 5:   **end if**
  - 6: **end for**
  - 7: **return**  $a$
-

2. Katso maxmin.java tiedosto.

3. (a) Jos meillä on  $n > 0$  kappaletta positiivisia reaalilukuja  $(a_1, a_2, \dots, a_n)$ , niiden aritmeettinen keskiarvo lasketaan oheisen kaavan perusteella:

$$\frac{1}{n} \sum_{i=1}^n a_i.$$

Algoritmi AriAve laskee syötteenä tulevan taulukon  $A$  alkioden aritmeettisen keskiarvon.

- (b) Jos meillä on  $n > 0$  kappaletta positiivisia reaalilukuja  $(a_1, a_2, \dots, a_n)$ , niiden harmoninen keskiarvo lasketaan oheisen kaavan perusteella:

$$\frac{n}{\sum_{i=1}^n \frac{1}{a_i}}.$$

Algoritmi HarAve laskee syötteenä tulevan taulukon  $A$  alkioden harmonisen keskiarvon.

- (c) Jos meillä on  $n > 0$  kappaletta positiivisia reaalilukuja  $(a_1, a_2, \dots, a_n)$ , niiden geometrinen keskiarvo lasketaan oheisen kaavan perusteella:

$$\sqrt[n]{a_1 a_2 \cdots a_n}.$$

Algoritmi GeoAve laskee syötteenä tulevan taulukon  $A$  alkioden geometrinen keskiarvon.

---

AriAve ( $A[0, 1, \dots, n-1]$ )

**alkuehto:**  $n > 0$

```

1: result  $\leftarrow 0$ 
2: sum  $\leftarrow 0$ 
3: for  $i \leftarrow 0$  to  $n-1$  do
4:   sum  $\leftarrow$  sum +  $A[i]$ 
5: end for
6: result  $\leftarrow$  sum/ $n$ 
7: return result
```

---

---

HarAve ( $A[0, 1, \dots, n - 1]$ )

**alkuehto:**  $n > 0$

1:  $result \leftarrow 0$

2:  $sum \leftarrow 0$

3: **for**  $i \leftarrow 0$  to  $n - 1$  **do**

4:    $sum \leftarrow sum + 1/A[i]$

5: **end for**

6:  $result \leftarrow n/sum$

7: **return**  $result$

---

---

GeoAve ( $A[0, 1, \dots, n - 1]$ )

**alkuehto:**  $n > 0$

1:  $result \leftarrow 0$

2:  $prod \leftarrow 1$

3: **for**  $i \leftarrow 0$  to  $n - 1$  **do**

4:    $prod \leftarrow prod \cdot A[i]$

5: **end for**

6:  $result \leftarrow \sqrt[n]{prod}$

7: **return**  $result$

---

4. Katso Averages.java tiedosto.

5. Tapoja on monia. Tässä on esitetty yksinkertainen, mutta hieman tehoton tapa. Se tarkistaa kullekin taulukon alkioille erikseen, onko kyseinen arvo taulukon moodi. Algoritmi palauttaa **viimeisenä löydetyn** moodin arvon.

---

FindMode ( $A[0, 1, \dots, n - 1]$ )

**alkuehto:**  $n > 0$

```
1:  $modePos \leftarrow 0$ 
2:  $modeCount \leftarrow 1$ 
3: for  $i \leftarrow 1$  to  $n - 1$  do
4:    $count \leftarrow 0$ 
5:   for  $j \leftarrow 0$  to  $n - 1$  do
6:     if  $A[j] = A[i]$  then
7:        $count \leftarrow count + 1$ 
8:     end if
9:   end for
10:  if  $count \geq modeCount$  then
11:     $modePos \leftarrow i$ 
12:     $modeCount \leftarrow count$ 
13:  end if
14: end for
15: return  $A[modePos]$ 
```

---

6. Tapoja on monia, esim. algoritmi 5. Taulukon lajitteluun voidaan käyttää esim. kuplalajittelua (algoritmi 3) tai valintalajittelua (algoritmi 4). Kumpikin niistä on tosin hidas. Tehokkaampia lajittelutapoja esitellään myöhemmin.

---

**Algorithm 3** Lajittelee taulukon  $A$  kuplalajittelulla. Jos taulukossa on keskenään väärässä järjestyksessä olevia, niin vaihdetaan niiden paikkaa. Kierroksen  $i$  lopuksi  $(i+1)$ :nneksi suurin alkio on viety lopulliselle paikalleen. Algoritmi voidaan lopettaa heti, kun vaihtoja ei enää tehdä. Kirjaimet  $T$  ja  $F$  tarkoittavat tässä loogisia vakioita true ja false.

---

BubbleSort ( $A[0, 1, \dots, n-1]$ )

```
1:  $i \leftarrow 0$ 
2: repeat
3:   vaihto  $\leftarrow F$ 
4:   for  $j \leftarrow 0$  to  $n-2-i$  do
5:     if  $A[j] > A[j+1]$  then
6:       vaihda  $A[j] \leftrightarrow A[j+1]$ 
7:       vaihto  $\leftarrow T$ 
8:     end if
9:   end for
10:   $i \leftarrow i+1$ 
11: until vaihto =  $F$ 
```

---

---

**Algorithm 4** Lajittelee taulukon  $A$  paikallaan toimivalla valintalajittelulla. Indeksissä  $m$  on kullakin kierroksella sisemmän silmukan suorituksen jälkeen taulukon lajittelemattoman osan (siis  $A[j], \dots, A[n-1]$ ) pienin alkio, joka vaihdetaan paikalleen.

---

SelectionSort ( $A[0, 1, \dots, n-1]$ )

```
1: for  $j \leftarrow 0$  to  $n-2$  do
2:    $m \leftarrow j$ 
3:   for  $i \leftarrow j+1$  to  $n-1$  do
4:     if  $A[i] < A[m]$  then
5:        $m \leftarrow i$ 
6:     end if
7:   end for
8:   vaihda  $A[m] \leftrightarrow A[j]$ 
9: end for
```

---

---

**Algorithm 5** Taulukon alkioiden mediaanin laskenta.

---

SelectMedian ( $A[0, \dots, n-1]$ )

```
1: BubbleSort( $A$ )
2: if  $n$  on pariton then
3:    $r \leftarrow A[(n-1)/2]$ .
4: else
5:    $r \leftarrow (A[n/2] + A[n/2-1])/2$ .
6: end if
7: return  $r$ 
```

---



7. Katso tiedostot Median.java ja Mode.java.

8. Katso Alphaorder.java tiedosto.