

811312A Tietorakenteet ja algoritmit 2018-2019, Harjoitus 2

Harjoituksen aiheena on algoritmien oikeellisuus.

Tehtävä 2.1 "Kahvipurkkiongelman". Kahvipurkissa P on valkoisia ja mustia kahvipapuja, yhteensä vähintään kaksi kappaletta. Lisäksi saatavilla on suuri määrä mustia papuja. Osoita, että seuraava algoritmi päättyy. Merkitään purkissa P kulloinkin olevien papujen määrää $N(P)$:llä.

```
Syöte: Kahvipurkki P, jossa on valkoisia ja mustia papuja,
alussa  $N(P) \geq 2$ .
Tuloste: Purkin viimeinen papu.

PAVUT(P)
1.  while  $N(P) > 1$ 
2.      valitse satunnaisesti kaksi papua purkista P
3.      if pavut samanväriset
4.          heitä pois pavut ja lisää purkkiin musta papu
5.      else
6.          palauta valkoinen papu purkkiin ja heitä pois musta papu
7.      return purkkiin jäänyt papu
```

Voidaanko viimeisen pavun väriä tietää ennakolta, jos tiedetään purkissa alussa olevien valkoisten ja mustien papujen lukumäärät?

Tehtävä 2.2 Seuraava ongelma tunnetaan **selkäreppuongelmana (knapsack problem)**: On annettu kokonaislukujen joukko $S = \{s_1, s_2, \dots, s_n\}$ ja kokonaisluku T. Etsi joukon S sellainen osajoukko (mikäli sellainen osajoukko on olemassa), sen alkioden summa on tarkalleen T. Esimerkiksi, kun $S = \{5, 10, 1, 2, 11\}$, niin jos $T=24$, on olemassa ratkaisu $\{1, 2, 10, 11\}$, mutta jos $T=25$, ratkaisua ei ole. Näytä, että kumpikaan seuraavista algoritmeista ei ole oikea ratkaisu selkäreppuongelmaan:

- Valitaan osajoukon alkiot joukosta S järjestyksessä pienimmästä suurimpaan (best-fit).
- Valitaan osajoukon alkiot joukosta S järjestyksessä suurimmasta pienimpään.

Tehtävä 2.3 Seuraavan algoritmin oletetaan laskevan syötteenä saamansa taulukon alkioden summan.

```
Syöte: Taulukko  $A[1, \dots, n]$ ,  $n \geq 1$ 
Tuloste: Taulukon alkioden summa

ALKIOIDEN_SUMMA(A)
1.  sum = 0
2.  i = 1
3.  while  $i \leq n$ 
4.      sum = sum + A[i]
5.      i = i+1
6.  return sum
```

Laske algoritmia käyttäen taulukon $A = \{3, 4, 2\}$ alkioden summa (vastaus: 9). Osoita algoritmi oikeaksi. Osoita algoritmin päättyminen sopivan konvergentin avulla. Muotoile ja todista oikeaksi silmukkainvariantti, joka vahvistaa algoritmin tuottavan oikean tuloksen.

Ohjelmointitehtävät toisella puolella ->

Ohjelmointitehtävät

Tehtävä 2.4 Seuraavan algoritmin tulisi kääntää syötetaulukon alkioden järjestys päinvastaiseksi:

```
Syöte: Taulukko  $A[0,1,\dots,n-1]$ ,  $n \geq 1$  (n siis taulukon koko)  
Tuloste: Kääntää taulukon A alkiot päinvastaiseen järjestykseen  
REVERSE(A,n)  
1.   i = 0  
2.   while i <= n/2  
3.       temp = A[i]  
4.       A[i] = A[n-i-1]  
5.       A[n-i-1] = temp  
6.       i = i+1  
7.   return
```

Algoritmi ei kuitenkaan ole korrekti. Implementoi algoritmi ja yritä löytää virhe. Korjaa tämän jälkeen ohjelma (ja algoritmi) toimimaan oikein.

Tehtävä 2.5 tehdään jos aikaa vielä riittää. Ns. **puolitushaun** avulla voidaan hakea annettu arvo taulukosta, jossa alkiot ovat suurusjärjestyksessä. Algoritmi toimii siten, että aluksi luetaan taulukon puolivälissä oleva arvo, jonka perusteella tiedetään, onko haettu arvo alku- vai loppupuoella taulukkoa. Tämän jälkeen puolitetaan jäljellä oleva osa, minkä jälkeen tiedetään missä taulukon neljänneksessä haettu arvo on. Näin jatkamalla löydetään arvo taulukosta tai havaitaan, että arvoa ei ole taulukossa. Seuraava iteratiivinen ohjelma pyrkii toteuttamaan puolitushaun järjestettyyn taulukkoon. Implementointi on kuitenkin virheellinen. Yritä löytää virhe. Implementoi ohjelma (joko C- tai Python-kielellä) ja yritä löytää testitapaus, jolla ohjelma toimii virheellisesti.

```
Syöte: Taulukko  $A[0,1,\dots,n-1]$ ,  $n \geq 1$ , taulukon alkiot ovat kasvavassa järjestyksessä  $A[0] \leq A[1] \leq \dots \leq A[n-1]$ . Luku  $x$  jota haetaan taulukosta.  
Tulostus: Alkion  $x$  indeksi taulukossa tai arvo -1, jos  $x$  ei esiinny taulukossa.  
HAKU(A,x)  
1. low=0  
2. up=n-1  
3. while low <= up  
4.     r =  $\lfloor (low + up) / 2 \rfloor$   
5.     if A[r] < x  
6.         low = r  
7.     else if A[r] > x  
8.         up = r  
9.     else  
10.        return r  
11. return -1
```

Merkintä $\lfloor x \rfloor$ ("floor") tarkoittaa suurinta kokonaislukua, joka on korkeintaan yhtäsuuri kuin x .