

Laskinten käyttö tentissä ei ole sallittua.

Huom: vastaa kysymyksiin selkeästi, hyvällä rakenteella ja napakasti.

1. Tarkastellaan oikealla olevaa Scala-ohjelmaa. Kuvaa yhdellä lauseella, millaisen laskennallisen ongelman metodi ratkaisee. Mitkä ovat metodin (i) pahimman ja (ii) parhaimman tapauksen ajoajat suhteessa syöteargumentin a pituuteen n ? Perustele jokainen vastauksesi korkeintaan muutamalla lauseella ja kerro, millaisilla syötteillä pahimman ja parhaimman tapauksen ajoajat ilmenevät.

```
def methodX(a: Array[Int], t: Int): Boolean = {  
  var i = 0  
  var found = false  
  while (i < a.length && !found) {  
    val v = a(i)  
    var j = i+1  
    while (j < a.length && !found) {  
      if (v * a(j) == t)  
        found = true  
      j += 1  
    }  
    i += 1  
  }  
  return found  
}
```

Kuinka ajoajat muuttuvat (vai muuttuvatko ollenkaan), jos ohjelman ensimmäinen rivi muutetaan muotoon “def methodX(a: List[Int], t: Int): Boolean = {”?
7 pistettä

2. Määrittele seuraavat käsitteet: (a) järjestämisalgoritmi, joka toimii paikallaan (*engl. in-place*) ja (b) stabiili järjestämisalgoritmi (*engl. stable sorting algorithm*).

Tarkastellaan oikeanpuoleista Scala-ohjelmaa. (i) Minkä yleisen järjestämisalgoritmin toteutus se on? (ii) Onko algoritmi stabiili? (iii) Toimiiko se paikallaan? (iv) Mikä on algoritmin pahimman tapauksen ajoaika? (v) Mikä on algoritmin parhaan tapauksen ajoaika? (vi) Muuttuvatko ajoajat, jos HELPER-metodin ensimmäisen rivin kommentit poistetaan?

Kysymyksissä (iv), (v) ja (vi) merkitse argumenttitaulukon a pituutta symbolilla n .

Perustele jokainen vastaus korkeintaan muutamalla lauseella.

```
def sort(a: Array[Int]): Unit = {  
  val aux = new Array[Int](a.length)  
  def helper(l: Int, m: Int, r: Int): Unit = {  
    // if (a(m-1) <= a(m)) return  
    var (i, j, d) = (l, m, l)  
    while (i < m && j <= r) {  
      if (a(i) <= a(j)) {aux(d) = a(i); i += 1}  
      else {aux(d) = a(j); j += 1}  
      d += 1  
    }  
    while (i < m) {aux(d) = a(i); i += 1; d += 1}  
    while (j <= r) {aux(d) = a(j); j += 1; d += 1}  
    d = l  
    while (d <= r) {a(d) = aux(d); d += 1}  
  }  
  def inner(l: Int, r: Int): Unit = {  
    if (l < r) {  
      val m = l + (r - l) / 2  
      inner(l, m)  
      inner(m+1, r)  
      helper(l, m+1, r)  
    }  
  }  
  inner(0, a.length-1)  
}
```

13 pistettä

3. Kerro, kuinka ketjutukseen (*engl. chaining, open hashing*) perustuvilla hajautustaulukoilla voidaan toteuttaa muuttuvatilainen tietorakenne joukoille. Kuvauksesi tulee määritellä käsitteet “hajautusfunktio”, “yhteentörmäys” ja “täyttöaste”, sekä vastata seuraaviin kysymyksiin: (i) Kuinka alkiodien lisääminen ja poistaminen toimivat? (ii) Mitä vaaditaan, että alkiodien lisääminen tapahtuu keskimäärin vakioajassa? (iii) Mitä “uudelleenhajautus” tarkoittaa ja miksi/

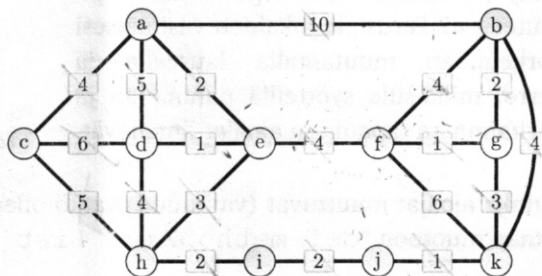
milloin se tulee suorittaa? Merkitse hajautustaulukon kokoa symbolilla m ja siihen lisättyjen alkoiden lukumäärää symbolilla n .

Tarkastellaan hajautustaulukkoa, jonka koko on $m = 7$, 32-bittisten kokonaislukujen tallettamista varten. Käytetään ketjutusta ja hajautusfunktiota $h(x) = x \bmod m$. Kuvaaja hajautustaulukon sisältö jokaisen askeleen jälkeen kun lisätään avaimet 15, 9, 8 ja 22 tässä järjestyksessä.

Kumpi seuraavista funktioista on luultavasti parempi hajautusfunktio merkkijonoille, $h_a(s) = \sum_{i=1}^{|s|} \text{int}(c_i) \bmod m$ vai $h_b(s) = \sum_{i=1}^{|s|} 101^{i-1} \times \text{int}(c_i) \bmod m$? Molemmissa funktioissa argumenttimerkkijono s sisältää $k = |s|$ merkkiä $c_1 \dots c_k$ ja $\text{int}(c)$ palauttaa merkin c kokonaislukuesityksen, mikä on tyyppillisesti välillä $[32, 126]$. Perustele vastauksesi, mahdollisesti käyttäen hyvin pientä esimerkkiä. 12 pistettä

4. Tarkastellaan yhtenäisiä ja painotettuja suuntaamattomia verkkoja. Määrittele käsite *minimaalinen virittävä puu* tällaisille verkoille. Kuvaile pseudokoodina, tai sanallisesti selkeällä ja rakenteellisella tavalla, joko (i) Kruskalin tai (ii) Primin algoritmi minimaalisten virittävien puiden löytämiseksi.

Millaisia tietorakenteita verkon esittäminen ja valitsemasi algoritmi tarvitsevat? Kuvaaja kuinka algoritmi toimii oikealla olevassa verkossa. Jos verkossa on n solmua ja m kaarta, niin mikä on algoritmin pahimman tapauksen ajoaika? Perustele vastauksesi korkeintaan muutamalla lauseella.



12 pistettä

5. Binäärimerkkijono on merkkijono, joka koostuu merkeistä 0 ja 1. Tarkastellaan seuraavaa ongelmaa: Annettuna positiivinen kokonaisluku n , kuinka monta erilaista n merkin pituista binäärimerkkijonoa, jossa ei ole kahta peräkkäistä merkkiä 1, on olemassa? Esimerkiksi arvolla $n = 3$ vastaus on 5 (merkkijonot ovat 000, 001, 010, 100 ja 101). Vastaavasti jos $n = 4$, niin vastaus on 8 (merkkijonot ovat 0000, 0001, 0010, 0100, 0101, 1000, 1001 ja 1010).

Kuvaaja dynaamiseen ohjelmointiin perustuva algoritmi ongelman ratkaisemiseksi. Anna sekä rekursiomäärittely että itse algoritmi (joko Scala-kielellä tai pseudokoodina). Algoritmisi tulisi toimia ajassa $O(n \log n)$. Näytä, kuinka algoritmisi ratkaisee ongelman kun $n = 10$ (älä yritä kirjoittaa kaikkia merkkijonoja, niitä on paljon). 8 pistettä

6. Tarkastellaan oikeanpuoleista rinnakaistettua Scala-ohjelmaa, joka laskee arvon $\text{op}(\dots \text{op}(\text{op}(a_0, a_1), a_2), \dots a_{n-1})$ taulukon $[a_0, a_1, \dots, a_{n-1}]$ arvoille jos binäärioperaatio op on liitännäinen (eli $\text{op}(x, \text{op}(y, z)) = \text{op}(\text{op}(x, y), z)$) kaikille arvoille x, y, z). Konstruktiio $\text{par.parallel}(\text{code1}, \text{code2})$ toimii kuten luentomateriaalissa eli suorittaa koodit code1 ja code2 rinnakkain ja palauttaa niiden paluuarvot.

```
def reduce[T](a: Array[T], op: (T,T)=>T): T = {
  require(a.nonEmpty)
  def inner(start: Int, end: Int): T = {
    if (start == end) a(start)
    else {
      val mid = start + (end - start)/2
      val (l, r) = par.parallel(
        inner(start, mid),
        inner(mid+1, end)
      )
      op(l, r)
    }
  }
  inner(0, a.length-1)
}
```

Mikä ovat ohjelman (i) kesto, (ii) työmäärä ja (iii) rinnakkaisuusaste? Kuinka ohjelman käytännön suorituskykyä voitaisiin parantaa? Miksi ohjelma ei aina toimi oikein jos op ei ole liitännäinen? Perustele jokainen vastauksesi lyhyesti. 7 pistettä

7. Mihin kellonaikaan lopetit tenttitehtäviin vastaamisen? 1 pistettä