```
 1
 2 public class mergesort {
 3     static int[] a;
 4     static int[] b;
 5
 6     public static void mergesort(int left, int right) {
 7         int i, j, k, m;
 8         System.out.println("mergesort("+left+", "+right+")");
 9         if (right > left)
10         {
11             m = (right + left) / 2;
12             mergesort(left, m);
13             mergesort(m + 1, right);
14             for (i = m; i >= left; i--)
15                 b[i]= a[i];
16             for (j = m + 1; j <= right; j++)
17                 b[right + m + 1 - j] = a[j];
18             i = left;
19             j = right;
20             for (k = left; k <= right; k++)
21                 if (b[i] < b[j])
22                 {
23                     a[k] = b[i];
24                     i = i + 1;
25                 }
26                 else
27                 {
28                     a[k] = b[j];
29                     j = j - 1;
30                 }
31         }
32     }
33
34     public static void main(String[] args) {
35         a = new int[6];
36         b = new int[6];
37
38         a[0] = 7; a[1] = 5; a[2] = 2; a[3] =4; a[4] =8; a[5] =3;
39         mergesort(0,5);
40     }
41
42 }
```

Teknillinen korkeakoulu TENTTI
Tietotekniikan laitos 30.10.2008
T-106.1220/1223 Tietorakenteet ja algoritmit T/Y

Kirjoita jokaiseen paperiin oma nimesi, oppilasnumerosi *tarkistusmerkkeineen*, tutkinto- tai koulutusohjelmasi, kurssikoodi ja kurssin nimi, päivämäärä, sali, palauttamiesi paperien lukumäärä sekä *allekirjoituksesi*.

1) Koodin analyysi (1p + 2p + 3p + 4p)

Liitteessä on annettu luokka `mergesort`, jossa on eräs toteutus *lomitusjärjestämisalgoritmille*. Tutustu koodiin huolella ja lue ensin kaikki kysymyskohdat. Vastaa sen jälkeen kaikkiin kysymyksiin. Tämä tehtävä on *tentin pakollinen osa*, josta on saatava vähintään 5p/10p, jotta loput tentistä tarkistetaan. Tämä tehtävä ei kuitenkaan yksistään riitä tentin läpäisyyn. Toisaalta viiteen pisteeseen ei edellytetä "täysin oikeaa vastausta" vaan oleellista on osoittaa *ymmärtäneensä* koodin toiminnan. Käytä siis aikaa myös perustelujen esittämiseen.

a) Järjestetääkö algoritmi alkiot pienimmästä suurimpaan vai päinvastoin? *Perustele*.
b) Onko algoritmi stabiili? *Perustele*.
c) *Mitä algoritmi tulostaa* (ks. rivi 8)?
d) *Selitä* metodin `mergesort` toiminta sanallisesti. *Anna* vähintään yksi *esimerkki*.

2) Terminologiaa (2p + 2p + 2p + 2p)

Määrittele seuraavat *käsitteet* (4 x 1p). *Anna* jokaisesta myös *esimerkki* (4 x 1p).

a) Abstrakti tietotyyppi (ADT)
b) Pino
c) Prioriteettijono
d) Binäärikeko

3) Hakurakenteet (4p + 4p)

Analysoi ja vertaile avoimeen osoitukseen perustuvia hajautusrakenteita ja tasapainotettuja hakupuita keskenään hakurakenteina.

a) *Perustele* ensin *neljä* (4) mielestäsi keskeisintä *vertailukriteeriä* tässä asiassa.
b) *Esitä* vertailun *tulokset* kunkin kriteerin suhteen.

Kirjoita vastauksesi siten, että kukin kriteeri perusteluineen on omana kappaleenaan ja vastaavasti vertailutulokset omina kappaleinaan.

4) Verkkoalgoritmit (4p + 4p)

a) *Selitä* joko Primin tai Dijsktran algoritmin toiminta sanallisesti. Käytä apuna sopivaa esimerkkiä.

b) Toimivatko em. algoritmit, jos painotetussa verkossa on negatiivisia särmiä? *Perustele* näkemyksesi molempien algoritmien osalta erikseen. Jos jompi kumpi algoritmi toimii mielestäsi virheellisesti, *anna esimerkki*.

Tekniska högskolan
Institutionen för datateknik
T-106.1220/1223 Datastrukturer och algoritmer T/Y

TENTAMEN
30.10.2008

Skriv på varje papper ditt namn, studentnummer *med kontrollbokstav* och examens- eller utbildningsprogram, kursens kod och namn, dagens datum, salen du är i, hur många papper du lämnar in och din *namnteckning*.

1) Analys av kod (1p + 2p + 3p + 4p)

Bilagan till tentamen innehåller `mergesort`-klassen och en implementation av mergesort-algoritmen. Bekanta dig noga med koden och läs först alla frågor. Besvara sedan varenda fråga. Denna uppgift är *tentamens obligatoriska del* som man måste få minst 5p/10p i för att resten av tentamen ska granskas. Att besvara bara den här uppgiften är inte tillräckligt för ett godkänt vitsord. Å andra sidan, för att få 5 poäng, behöver du inte svara "helt rätt"; det väsentliga är att *visa att du förstått* hur koden fungerar. Motivera därför ditt svar grundligt.

a) Hur ordnar den här algoritmen data: i stigande eller minskande ordning? *Motivera.*
b) Är den här algoritmen stabil? *Motivera.*
c) *Vad skriver den här algoritmen ut* (se rad 8)?
d) *Förklara* hur metoden `mergesort` fungerar. Ge minst ett exempel.

2) Terminologi (2p + 2p + 2p + 2p)

Definiera följande begrepp (4 x 1p). *Ge exempel* på varje begrepp (4 x 1p).

a) Abstrakt datatyp (ADT)
b) Stack
c) Prioritetskö
d) Binär heap

3) Sökstrukturer (4p + 4p)

Analysera och jämför balanserade sökträd och hashtabeller baserade på öppen adressering som sökstrukturer.

a) *Ange* de *fyra* (4) viktigaste *kriterierna* för jämförelsen. Motivera!

b) *Skriv resultatet* av jämförelsen enligt varje kriterium.

Behandla varje kriterium i ett skilt stycke (för att underlätta vitsordsgivningen). Likaledes, skriv jämförelseresultatet för varje kriterium i ett skilt stycke.

4) Grafalgoritmer (4p + 4p)

a) *Förklara* antingen Prims eller Dijkstras algoritm. Ge ett lämpligt exempel.

b) Hur fungerar algoritmerna om en graf har negativa kanter? *Motivera* din åsikt i båda fallen. Om någon av algoritmerna fungerar felaktigt, *ge ett exempel* på det.

Helsinki University of Technology
Department of Computer Science and Engineering
T-106.1220/1223 Data Structures and Algorithms T/Y

EXAMINATION
30.10.2008

Write on each paper your name, student number *with the control letter*, degree programme, and the course code and name. Also write the date, hall, the number of papers you return, and your *signature*.

1) Code analysis (1p + 2p + 3p + 4p)

In the attached listing, you can see the `mergesort` class that includes an implementation of the *merge sort algorithm*. Familiarize yourself thoroughly with the code and read all the questions first. After this, answer all the questions. This is the *compulsory part of the examination*, and you need at least 5p/10p in order to get the rest of the examination graded. However, this part alone is not enough to pass the examination. You do not need to give "perfectly correct" answers to get 5 points, but merely show that you have *understood* the functioning of the code. Thus, pay attention to your argumentation.

a) Does the algorithm sort the items in ascending or descending order? *Justify*.
b) Is this algorithm stable? *Justify*.
c) *What's the output of the algorithm* (see line 8)?
d) *Explain* how the `mergesort` method works. *Give* at least one *example*.

2) Terminology (2p + 2p + 2p + 2p)

Define the following *concepts* (4 x 1p). *Give an example* of each (4 x 1p).

a) Abstract Data Type
b) Stack
c) Priority Queue
d) Binary Heap

3) Search structures (4p + 4p)

Analyse and compare hashing methods based on open addressing with balanced search trees as search structures (dictionaries).

a) *Justify* the *four* (4) in your opinion most important *comparison criteria* for this.
b) *Present the results* of the comparison for each criterion.

Write your answer so that each criterion is in a separate paragraph together with the argumentation for it and the comparison results are likewise in paragraphs of their own.

4) Graph algorithms (4p + 4p)

a) *Explain* how either Prim's or Dijkstra's algorithm works. *Give* a suitable *example*.

b) Do the above algorithms work correctly if the graph has negative edges? *Argue* your opinion for each algorithm. If an algorithm does not work, *give an example* which shows this.