

TDDC91
Datastrukturer och algoritmer
Datortentamen (DAT1)
2017-10-23, 14–18

Examinator:	Erik Nilsson
Jour:	Magnus Nielsen (telefon 073-0822614).
Antal uppgifter:	7, OpenDSA inräknat.
Max poäng:	40 poäng
Preliminära gränser:	3 = 20p, 4 = 27p, betyg 5 = 35p (Kan komma att justeras)
Hjälpmedel:	Inga hjälpmedel tillåtna!
Bonuspoäng:	Eventuella intjänade bonuspoäng kommer att tillgodoräknas efter att gränsen för 3/G uppnåtts.

VÄNLIGEN IAKTTAG FÖLJANDE

- Du får själv välja om du vill skriva din lösning på papper eller på dator.
- Lösningar till olika problem skall placeras enkelsidigt på separata blad, eller i egen fil. Skriv inte två lösningar på samma papper eller i samma fil. Delproblem får dela papper / fil.
- Om inte annat framgår ska indexering av arrayer / listor börja från 0.
- Papper: Sortera lösningarna innan de lämnas in.
- Filer: Skicka in som lösning till rätt problem i tentaklienten, och döp filen till ett passande namn (exempelvis uppg1.txt, eller uppg1.odt).
- MOTIVERA DINA SVAR ORDENTLIGT: avsaknad av, eller otillräckliga, förklaringar resulterar i poängavdrag. Även felaktiga svar kan ge poäng om de är korrekt motiverade.
- Om ett problem medger flera olika lösningar, t.ex. algoritmer med olika tidskomplexitet, ger endast optimala lösningar maximalt antal poäng.
- Papper: Lämna plats för kommentarer.
- SE TILL ATT DINA LÖSNINGAR/SVAR ÄR LÄSLIGA.

Lycka till!

1. OpenDSA

(12 p)

Efter inloggning i datortentasytemet finns i startmenyn för Linux Mint:

- “OpenDSA” öppnar URL för tentamensversion av OpenDSA i Chromium

Logga in på OpenDSA med ditt tilldelade SC-nummer, och använd ditt personnummer som lösenord. SC-numret kan du alltid läsa av från din tenta live-klient som öppnades när du loggade in på datorn. Numret står i fältet för KlientID. Håll din klient öppen, så att du snabbt och smidigt kan ställa frågor till jourhavande lärare, eller skicka in uppgifter.

Genom att klicka på ditt inloggningsnamn kan du hela tiden se i betygsboken hur många av de poänggivande uppgifterna du löst hittills. När du har full poäng i betygsboken är du färdig med den här uppgiften och kan logga ut. På den här uppgiften behöver du inte skicka in något via tentamenssystemet.

2. Sortering

(6 p)

Efter en mycket lyckad karriär som DALG-student har du startat ett eget konsultföretag som erbjuder effektiva databehandlingslösningar åt storföretag. Ett av kundföretagen är ett finansbolag, Mobsters Inc., som har en stor kunddatabas. Då bolaget har gott rykte (missnöjda kunder brukar försvinna spårlöst) samt har funnits länge är databasen med kunder mycket stor, och dessutom växer den konstant. De har bestämt sig för ett nytt bonussystem, då de vill locka till sig alla potentiella kunder i hela världen, och de vill dessutom att du ska implementera det mot ett mycket generöst arvode. Nyligen utexaminerad tycker du självfallet att det låter fantastiskt.

Varje gång de gör vinst på en affär (vilket kan vara många gånger per dag, då de är mycket motiverade och bra på att övertala folk) ger de en bonus till kunderna. Ju längre man har varit kund, desto större bonus får man. Det finns en algoritm som räknar ut hur mycket den första kunden ska få baserat på hur många kunder det finns. Samma algoritm körs för kund nummer två, tre, fyra ... N (där N representerar antalet kunder) där kunderna får lite mindre per steg, fram till den N 'te kunden som får den minsta summan (det som är kvar).

Varje kund har ett heltal som representerar dess kund ID (första kunden har ID 1, andra kunden ID 2, osv.), datumobjekt som visar vilket datum de anslöt sig, strängar som representerar namn, efternamn samt kontoinformation. Tyvärr använder bolaget en väldigt begränsad databas som bara ger ett fullständigt utdrag ur kundregistret utan någon ordning. Du anser att listan behöver sorteras för att förenkla och effektivisera arbetet, och väljer att sortera efter ID (heltal).

- (a) Vilken eller vilka sorteringsalgoritmer väljer du att använda? Motivera. (3)
- (b) Vilken tidskomplexitet har din sorteringslösning? (1)
- (c) Antag att vi inte har tillgång till ID. Hur förändras sorteringsituationen? Vilka nya avvägningar och val gör du? (2)

3. Sorteringsalgoritmer

(2 p)

Sorteringsalgoritmer. Svaren behöver ej motiveras.

Givet följande osorterade array:

23	23	75	96	93	93	90	23	61	90	72	60	13	80	83
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Efter några iterationer av några olika sorteringsalgoritmer (iteration i bemärkelse fullständig körning av inre loop, alternativt rekursivt anrop) har vi dessa resultat:

A	23	23	23	60	13	61	72	75	80	83	90	93	93	96	90
B	23	23	75	96	93	93	90	23	61	90	72	60	13	80	83
C	13	23	23	23	60	75	96	93	93	90	61	72	90	80	83
D	13	23	23	96	93	93	90	75	61	90	72	60	23	80	83

Matcha halvsorterad array mot en algoritm. Felaktig matchning ger minuspoäng, dock kan uppgiften ej ge total minuspoäng. Ex: 1 korrekt matchning och 3 felaktiga ger 0 p. 2 korrekta ger 1p. 3 korrekta och 1 felaktig ger 1p, osv.

- (a) Bubblesort, med nedbubbling av minsta element
- (b) Insertionsort
- (c) Quicksort, elementet längst till höger i partitionen används som pivot
- (d) Selectionsort

4. Hashtabeller

(4 p)

Vi har en hashtabell med linjär adressering, med några element instoppade. Hash-funktionen är $h(x) = x \bmod \text{size}$.

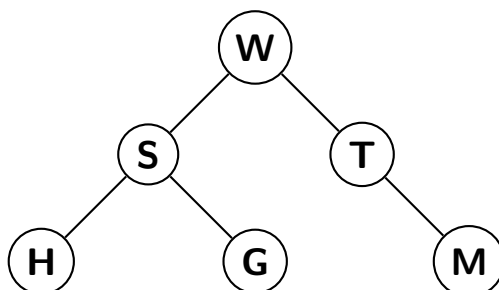
30	Null	2	23	13	5	26	27	Null	9
----	------	---	----	----	---	----	----	------	---

- (a) I vilken ordning har elementen stoppats in? Det finns flera lösningar. Ge fyra korrekta lösningar för maxpoäng. (2)
- (b) Om vi tar bort 23, hur kommer tabellen se ut? Det finns flera olika lösningar. Ge två olika, och förklara dem, för maxpoäng (2)

5. **Träd**

(6 p)

Studera följande träd:

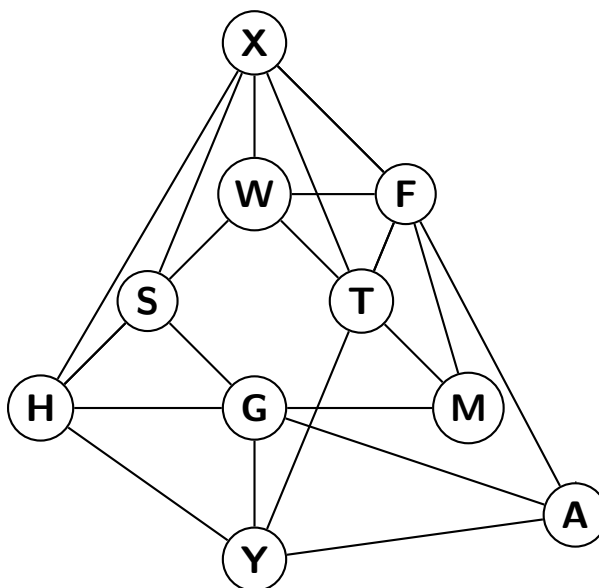


- (a) Är trädet ett binärt sökträd? Motivera! (2)
- (b) Är trädet balanserat? Motivera! (2)
- (c) Är trädet en min-heap, max-heap eller ingetdera? Motivera! (2)

6. **Graftraversering**

(4 p)

Studera följande graf:



Det räcker med en korrekt lösning på a) och b). Du behöver alltså inte räkna upp flera tänkbara lösningar.

- (a) I vilken ordning kommer noderna att traverseras med bredden först sökning, om vi utgår från A? (2)
- (b) I vilken ordning kommer noderna att traverseras med djupet först sökning, om vi utgår från A? (2)

7. Algoritmer och Tidskomplexitet

(6 p)

(a) Beskriv hur algoritmen heap sort fungerar.

(3)

(b) Beräkna tidskomplexiteten med avseende på n för följande funktion:

(1)

```
public int calc1(n) {
    res = 0;
    for (int i = 0; i < n; i++) {
        res++;
        for (int j = 0; j < n; j++) {
            res = res + 5;
        }
    }
    return res;
}
```

(c) Beräkna tidskomplexiteten med avseende på n för följande funktion:

(1)

```
public int calc2(n) {
    res = 0;
    for (int i = n; i > 0; i=i/2) {
        res++;
    }
    return res;
}
```

(d) Beräkna tidskomplexiteten med avseende på n för följande funktion:

(1)

```
public int calc3(n) {
    res = 0;
    for (int i = n; i > 0; i--) {
        res += calc2(i);
    }
    return res;
}
```