

TDDI16
Datastrukturer och algoritmer
Datortentamen (DAT1)
2018-10-29, 14–18

Examinator:	Erik Nilsson
Jour:	Filip Strömbäck (telefon 013-28 27 52).
Antal uppgifter:	8, OpenDSA inräknat.
Max poäng:	40 poäng
Preliminära gränser:	Betyg 5 = 35p, 4 = 27p, 3 = 20p.
Hjälpmedel:	Inga hjälpmedel tillåtna!

VÄNLIGEN IAKTTAG FÖLJANDE

- Observera att betygsgränserna kan komma att justeras.
- 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).
- **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 datortentasystemet 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. Sorteringsalgoritmer

(2 p)

Svaren behöver ej motiveras.

Efter **ett fåtal** iterationer av några olika sorteringsalgoritmer på den osorterade arrayen *Original* i tabellen nedan (iteration i bemärkelse fullständig körning av inre loop, alternativt rekursivt anrop) har vi resultaten i 1, 2, 3 och 4.

Original:	30	92	2	55	5	91	41	99	26	68	22	8	0	72	25
1:	2	30	55	92	5	91	41	99	26	68	22	8	0	72	25
2:	72	68	41	55	25	8	0	30	26	5	22	2	91	92	99
3:	0	2	5	55	92	91	41	99	26	68	22	8	30	72	25
4:	0	2	5	22	8	25	41	72	26	68	55	30	91	99	92
Sorterad:	0	2	5	8	22	25	26	30	41	55	68	72	91	92	99

Matcha delvis sorterad array mot en algoritm. Felaktig matchning ger minuspoäng, dock kan uppgiften ej ge total minuspoäng. För full poäng krävs 4 korrekta svar.

- Insertionsort
- Heapsort
- Selectionsort
- Quicksort, elementet längst till höger i partitionen används som pivot

3. Para ihop strumpor

(2 p)

Du har just tvättat alla dina strumpor. Som vanligt har några försvunnit i tvätten, så du står inför det vanliga problemet att para ihop de strumporna som inte har kommit bort. Tidigare har du helt enkelt parat ihop strumporna på måfå, men det har lett till dömande blickar från dina modemedvetna kompisar. Därför vill du den här gången även ta hänsyn till strumpornas färg!

Beskriv hur du effektivt kan komma fram till hur många par strumpor du har kvar efter tvätten. Varje par måste vara i samma färg. Beskriv vilken datarepresentation du använder, men välj själv en lämplig representation. Notera även vad din lösning har för tidskomplexitet, där n är antalet strumpor.

4. Den rättvisa festen

(4 p)

Du arrangerar en fest för dina kompisar. Det finns många olika typer av drycker: Coca-cola, kaffe, te... vatten... etc. Framåt kvällen framkommer det att folk har druckit olika mycket av olika saker, vilket självfallet orsakar missnöje bland gästerna! Som snäll värd vill du självfallet lösa problemet på ett smidigt sätt.

För att lösa problemet har du baserat på dina inventarier räknat ut hur mycket av varje dryck var och en av gästerna kan få så att alla kan få lika mycket. Dessutom har du sammanställt en lista av vad var och en av gästerna har druckit. Dessvärre är listorna i oordning eftersom konsumtionen har skett i olika ordning (och minnet sviktar, inte alla hanterar koffein så bra...).

Det enda som är kvar för att alla ska bli nöjda är att dela ut de drycker som är kvar för att alla ska uppnå sin rättvisekvota. Du har därför bett gästerna att ställa sig i kö för att de ska få sin beskärda del av drycken. För att inte skapa ytterligare missnöje vill du hitta ett sätt att snabbt lista ut vilka drycker du ska ge till personen som står längst fram i kön för att denne ska bli nöjd. Antag att du har två listor tillgängliga för att lösa problemet, dels listan av vilka drycker varje gäst borde ha fått samt listan av vad den nuvarande gästen har konsumerat. Exempelvis:

Rättvisekvota: kaffe te kaffe te Coca-cola Pepsi-cola vatten vatten
Konsumerat: Coca-cola kaffe vatten te kaffe

I detta fallet borde gästen få: te, vatten och Pepsi-cola.

Notera: Du kan inte anta att varken rättvisekvotan eller listan över konsumerade drycker är sorterade. Du har också fått i dig en rejäl dos koffein...

Notera: Du kan anta att ingen gäst har fått mer än sin rättvisekvota.

- (a) Beskriv din lösning. Motivera dina (ditt) val! (3)
- (b) Vilken tidskomplexitet har din lösning? Alltså, tidskomplexiteten för dig att komma fram till vad en specifik gäst ska få (inte samtliga). Antag att n är det maximala antalet drycker som någon har druckit. (1)

5. Hashtabeller

(4 p)

Vi har en hashtabell med linjär adressering och några element instoppade. Hashfunktionen är $h(x) = x \bmod \text{size}$ (arrayindex står under arrayen för tydlighets skull).

0	20	null	13	null	25	6	7	18	26
0	1	2	3	4	5	6	7	8	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 18 från den delvis fyllda hashtabellen ovan (remove / delete), hur kommer tabellen se ut? Det finns flera olika lösningar. Ge två olika lösningar som inte hashar om hela tabellen utom möjligen i värsta fallet, och förklara dem, för maxpoäng. (2)

6. Algoritmer och Tidskomplexitet (6 p)

(a) Beräkna tidskomplexiteten med avseende på n för följande funktion: (1)

```
int fun(n) {
    int res = 0;
    for (int i = 1; i < 2*n; ++i) {
        res++;
    }
    return res;
}
```

(b) Beräkna tidskomplexiteten med avseende på n för följande funktion: (1)

```
int more_fun(n) {
    int res = 0;
    for (int i = n; i >= 1; i=i/2) {
        res += i;
    }
    return res;
}
```

(c) Beräkna tidskomplexiteten med avseende på n för följande funktion: (2)

```
int super_fun(n) {
    int res = 0;
    for (int i = 3; i < n; ++i) {
        res += fun(2*i) + more_fun(i);
    }
    return res;
}
```

(d) Du har skrivit ett program som ritar ut kvadrater i terminalen m.h.a ASCII-tecken. (2)

Programmet frågar först efter en storlek och skriver sedan ut kvadraten med storlek 1, 2 osv. till och med storleken som matades in. Beskriv med hjälp av ett uttryck i Ordo-notation hur många tecken som skrivs ut om talet n matas in. Du behöver inte ta hänsyn till nyradstecken eller inmatning.

Exempel med inmatning av talet 3:

```
+---+
|  |
+---+
+-----+
|   |   |
|   |   |
+-----+
+-----+
|       |
|       |
|       |
+-----+
```

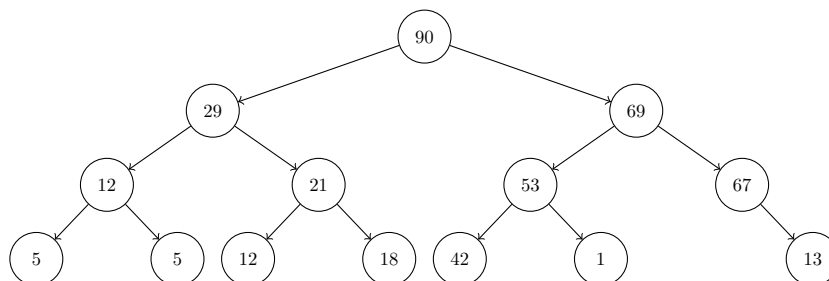
Tips: Om du tycker att det känns konstigt att räkna tecken, tänk dig att utskrift av ett tecken tar konstant tid och räkna sedan den totala tiden som utskriften tar. Det är ekvivalent med antalet tecken som skrivs ut.

Tips: Skriv kod som löser problemet och analysera den!

7. Träd

(6 p)

Studera följande träd:

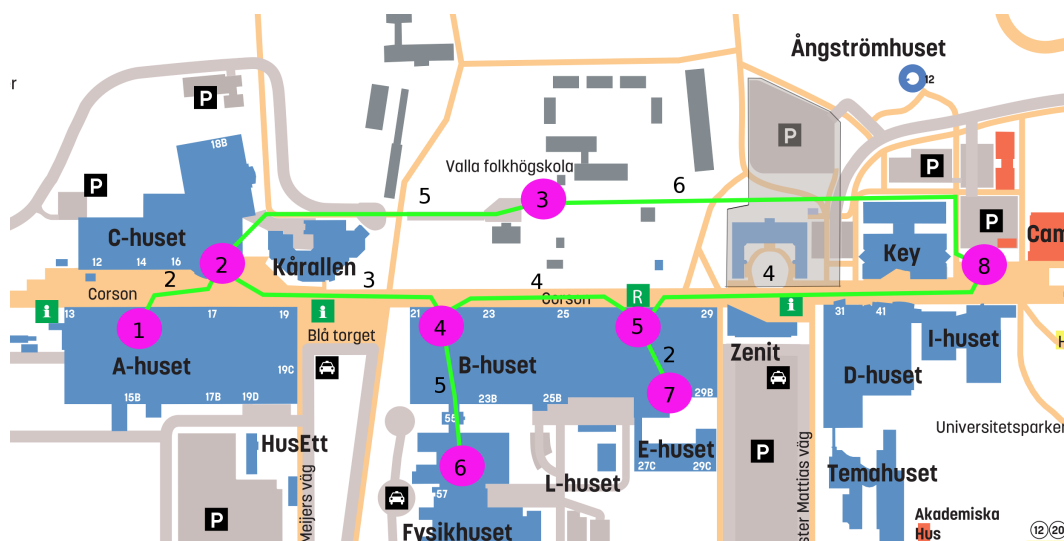


- Är trädet ett binärt sökträd? Motivera! (2)
- Är trädet balanserat? Motivera! (2)
- Är trädet en min-heap, max-heap eller ingetdera? Motivera! (2)

8. Grafteraversering

(4 p)

Studera följande viktade graf som beskriver campus:



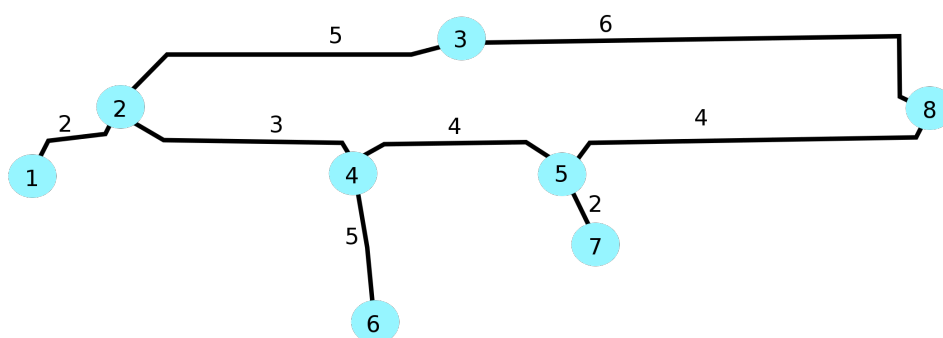
(en tydligare version av samma graf finns på sista sidan i tentan)

- Du pluggar DALG i C-huset (nod 2), när du plötsligt blir överkommen av hunger och yrsel. Omedelbart slår det dig att du, i ren extas, glömt att äta på några dygn och att du snabbt måste fylla på med falafelenergi (nod 8). Gör en breddenförstöring för att hitta kortaste vägen innan yrseln blir för svår! Redovisa vilka noder som besöks. Varför ger det inte nödvändigtvis kortaste vägen? (2)
- Nästa dag händer nästan samma sak, men eftersom du faktiskt åt igår är du piggare och beslutar dig för att göra en genomkörning av Dijkstras algoritmen och således försäkra dig om att du har den kortaste vägen. Redovisa vilken ordning noderna besöks och vilken väg du väljer. (1)
- Du beslutar dig för att du vill dela med dig av din nyfunna falafelenergi med din kära kursledare, och måste därmed passera B-huset (nod 7) på väg tillbaka till C-huset. Hur kan du använda Dijkstras för att hitta rätt väg? (1)

9. Bonusuppgifter i Kattis

(0-4 p)

Eventuella bonuspoäng från extrauppgifterna i Kattis. På den här uppgiften behöver du inte skicka in något via tentasystemet. Poängen förs in i efterhand och räknas endast mot högre betyg.



Figur 1: Högkontrastversion av kartan till uppgift 8