

Intro

Uppgift	Uppgiftstitel	Totalpoäng	Uppgiftstyp
i	Information		Dokument

Terminologi

Uppgift	Uppgiftstitel	Totalpoäng	Uppgiftstyp
1.1	Terminologi	10	Matchning

Datakomprimering

Uppgift	Uppgiftstitel	Totalpoäng	Uppgiftstyp
2.1	Huffman-kodning	6	Sifferfält
2.2	Huffman-avkodning	6	Textfält

Hashtabell

Uppgift	Uppgiftstitel	Totalpoäng	Uppgiftstyp
3.1	Hashtabell	10	Textalternativ

Grafalgoritmer

Uppgift	Uppgiftstitel	Totalpoäng	Uppgiftstyp
4.1	Minsta uppspännande träd	3	Flersvarsfråga
4.2	Graf-traversering	3	Flervalsfråga

Träd

Uppgift	Uppgiftstitel	Totalpoäng	Uppgiftstyp
5.1	Traversering av binärt träd	12	Sammanfatt

5.2	Traversering av ordnat träd	2	Matchning
-----	-----------------------------	---	-----------

Pseudokod

Uppgift	Uppgiftstitel	Totalpoäng	Uppgiftstyp
6.1	Kopiering Fält av Riktad Lista	12	Programmering

Egenskaper för datatyper

Uppgift	Uppgiftstitel	Totalpoäng	Uppgiftstyp
7.1	Ordning	6	Matchning
7.2	Sortering	6	Matchning

Komplexitet

Uppgift	Uppgiftstitel	Totalpoäng	Uppgiftstyp
8.1	Komplexitet för algoritm	8	Sammansatt
8.2	Ordo	6	Sammansatt

Sortering

Uppgift	Uppgiftstitel	Totalpoäng	Uppgiftstyp
9.1	Komplexitet för sorteringsalgoritmer	3	Matchning
9.2	Stabila/rekursiva algoritmer	3	Flersvarsfråga
9.3	Bubbelsortering	4	Textfält

Kommentarer

Uppgift	Uppgiftstitel	Totalpoäng	Uppgiftstyp
10.1	Förtydliganden (inga poäng)	0	Essä

i Information

- Försök på alla uppgifter! Uppgifterna är inte ordnade på något speciellt sätt.
- Det är ditt ansvar att övertyga oss att du besitter den kunskap som efterfrågas.
- Det är viktigt att du löser den *givna* uppgiften!
- Du har tillgång till en miniräknarfunktion i Inspira om du behöver.
- På sista "frågan" i denna tentamen finns plats att skriva kommentarer om du vill förtydliga något kring dina svar på någon fråga som inte har textsvar. "Frågan" ger inga poäng.
- Endast de som är underkända på tidigare tentamina får skriva denna tentamen. Plussning (dvs att skriva om efter du blivit godkänd i syfte att få högre betyg) är inte tillåtet.
- Ola kommer att besöka tentasalen under provets gång, c:a 10.00.

Betyg:

Maximal poäng är 100.

- För betyg **3** (godkänt) krävs **50** poäng.
- För betyg **4** krävs **65** poäng.
- För betyg **5** krävs **80** poäng.

Lycka till!

Ola, Niclas, Nina

1 Terminologi

Nedan kommer 10 sant/falskt frågor. Varje rätt svar ger 1p. Felaktigt eller inget svar alls ger 0p på den frågan. Med datatyp avses en abstrakt datatyp med definition enligt boken.

	Falsk	Sant
En sammansatt datatyp kan vara homogen.	<input type="radio"/>	<input checked="" type="radio"/>
En datatyp där man beskrivit både representationen för objekten och utförandet av operationerna kallas sammansatt datatyp.	<input checked="" type="radio"/>	<input type="radio"/>
I en sammanhängande (connected) graf är alla noder grannar till alla andra.	<input checked="" type="radio"/>	<input type="radio"/>
Mängd är ett exempel på en heterogen datatyp.	<input checked="" type="radio"/>	<input type="radio"/>
Ett riktat träd måste också vara ordnat	<input checked="" type="radio"/>	<input type="radio"/>
Elementen i en oordnad datatyp kan sorteras.	<input checked="" type="radio"/>	<input type="radio"/>
En konkret datatyp kan vara implementerad.	<input type="radio"/>	<input checked="" type="radio"/>
En datatyp vars objekt inte består av element som i sin tur är datatypsobjekt kallas enkel datatyp. Tex heltal.	<input type="radio"/>	<input checked="" type="radio"/>
Listorna ["ett", "tre"] och [(1,2,3) , (4,5,6)] har samma struktur.	<input type="radio"/>	<input checked="" type="radio"/>
Man kan prata om en array som innehåller två element och tre värden.	<input checked="" type="radio"/>	<input type="radio"/>

Totalpoäng: 10

2.1 Huffman-kodning

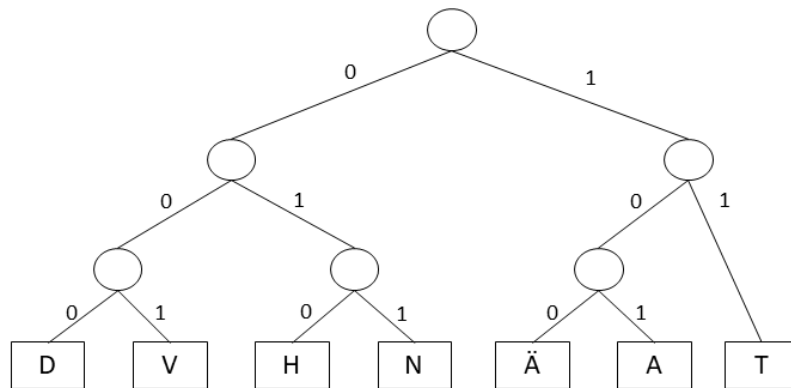
Givet ett Huffmanträd för strängen MISSISSIPPI:

- a) Hur lång blir den längsta sekvensen bitar som behövs för att koda ett tecken? (3)
- b) Hur lång blir den kortaste sekvensen bitar som behövs för att koda ett tecken? (1)

Totalpoäng: 6

2.2 Huffman-avkodning

Du har fått ett mail som innehåller tips mot spridning av virus. Informationen är komprimerad med ett Huffmanträd där nollor motsvarar vänster och ettor höger (se figur).



Vad står det i meddelandet?

110011001111010101011000

Svar: (TVÄTTAHAND)

Totalpoäng: 6

3 Hashtabell

För varje beskrivning, ange vilken teknik som avses (2p per rätt svar):

a) Om ett element kolliderar (platsen redan upptagen) sätter man in det på den första lediga

platsen efter den upptagna. (Sluten hashing, kvadratisk teknik, Öppen hashing, **Sluten hashing, linjär teknik**)

b) Om ett element kolliderar (platsen redan upptagen) sätter man först in det på nästa plats, sedan fyra platser fram sedan nio fram etc, dvs $H, H+1, H+4, \dots, H+i^2, \dots$ där H är elementets

hashvärde. (Sluten hashing, linjär teknik, **Sluten hashing, kvadratisk teknik**, Öppen hashing)

c) Vi använder oss av öppen hashing och har en vektor som är k element stor. Hur många element n bör man som mest sätta in i hashtabellen för att fortfarande ha effektiva operationer?

(0.5k, k, **2k**, $\log(k)$, $n \bmod k$)

d) Elementen 1, 8, 27, 64 och 125 har satts in i en tabell med hashfunktionen $h(x) = x \bmod 7$. Resultatet blev följande:

index	0	1	2	3	4	5	6
värde	125	1	8			64	27

Vilken teknik har använts? (**Sluten hashning, kvadratisk teknik**, Sluten hashning, linjär teknik, Öppen hashning)

e) Hur definieras en hashtabells fyllnadsgrad λ ? (Antal insatta element modulus antal platser i tabellen, Kvoten mellan antalet platser i tabellen och antalet insatta element., **Kvoten mellan antalet insatta element och antalet platser i tabellen.**, Kvoten mellan antalet insatta och borttagna element i tabellen.)

Totalpoäng: 10

4.1 Minsta uppspännande träd

Vilken/vilka av följande algoritmer kan användas för att ta fram ett minsta uppspännande träd för en viktad graf? Du får poäng för rätt svar och avdrag för fel svar. Minsta poäng är 0p

Välj ett eller flera alternativ

☐ Floyds

☐ Kruskals



☐ Prims



☐ Dijkstras

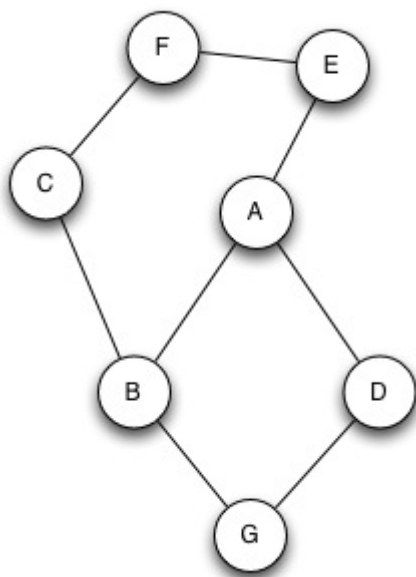
Totalpoäng: 3

4.2 Graf-traversering

Vilket sätt att traversera grafen nedan har använts om ordningen som noderna besöks blir C-B-G-D-A-E-F?

Välj ett alternativ:

- ☐ Inget av de övriga svaren
- ☐ Det går inte att säga
- ☐ Bredden först
- ☐ Djupet först



Totalpoäng: 3

5.1 Traversering av binärt träd

Illustrationen till vänster visar ett träd. Betrakta trädet som ett **binärt** träd. Nedanstående frågor handlar om traversering av trädet. I svaren, skriv **nodernas etiketter, separerade med kommatecken**, t.ex. "A,B,C,D,E,F,G".

a) I vilken ordning kommer noderna att besökas om trädet traverseras enligt **bredden-först**?

(X,A,Z,Y,C,M,P)

b) I vilken ordning kommer noderna att besökas om trädet traverseras enligt **djupet-först, preorder**?

(X,A,Y,Z,C,P,M)

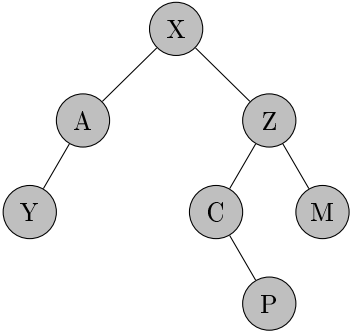
c) I vilken ordning kommer noderna att besökas om trädet traverseras enligt **djupet-först, inorder**?

(Y,A,X,C,P,Z,M)

d) I vilken ordning kommer noderna att besökas om trädet traverseras enligt **djupet-först, postorder**?

(Y,A,P,C,M,Z,X)

Totalpoäng: 12



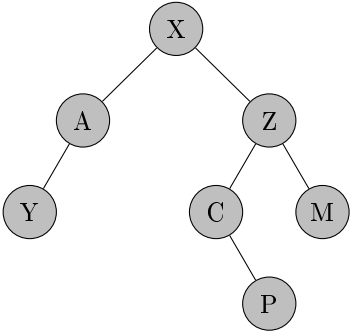
5.2 Traversering av ordnat träd

Illustrationen till vänster är densamma som för föregående uppgift. Ange nedan för var och en av traverseringsordningarna om dom skulle ge *samma* nodsekvens eller en *förändrad* nodsekvens om vi betraktade trädet som ett **ordnat** träd i stället för ett **binärt** träd.

Matcha ihop värdena:

	Samma	Förändrad
Bredden-först	<input type="radio"/> ✓	<input type="radio"/>
Djupet-först, preorder	<input type="radio"/> ✓	<input type="radio"/>
Djupet-först, inorder	<input type="radio"/>	<input type="radio"/> ✓
Djupet-först, inorder	<input type="radio"/> ✓	<input type="radio"/>

Totalpoäng: 2



6 Kopiering Fält av Riktad Lista

Skriv pseudokod för en funktion som skapar och returnerar en kopia av ett objekt av typen Fält av Riktad lista. Du får bara använda funktioner i gränsytan till datatyperna. Du ser gränsytorna till vänster. Du får anta att alla positioner i fältet innehåller en lista med noll eller flera element. Kopieringen ska innehålla alla elementvärden i samma ordning.

Algoritmen ska ha följande huvud:

Algorithm CopyArrayOfDList(a: Array)

Det är möjligt att *modularisera* lösningen, dvs. att identifiera en eller flera delalgoritmer/funktioner och beskriva dem separat. Ge dom i så fall lämpliga namn och kommentarer. Speciellt viktigt är kommentarer som beskriver inparametrar och returvärden. Tänk på att funktioner i pseudokod kan returnera flera parametrar.

Det är inte ett krav att modularisera lösningen, men det krävs för att ge fulla poäng.

Det som kommer att bedömas är

- att pseudokoden löser uppgiften under de givna förutsättningarna,
- att pseudokoden är fri från språkspecifika konstruktioner (inga **i++** eller **for i in range(...)**, etc.),
- att koden är korrekt indenterad,
- att koden är rimligt kommenterad,
- om koden är modulariserad, och
- om koden har optimal komplexitet ($g(n)$ är viktigast).

Skriv din kod här:

1	
---	--

```
abstract datatype Array(val, index)
  Create(lo, hi: index) → Array(val, index)
  Low(a: Array(val, index)) → index
  High(a: Array(val, index)) → index
  Set-value(i: index, v: val, a: Array(val, index))
    → Array(val, index)
  Has-value(i: index, a: Array(val, index)) → Bool
  Inspect-value(i: index, a: Array(val, index)) → val
```

```
abstract datatype DList(val)
```

```
auxiliary pos
```

```
  Empty() → DList(val)
  Isempy(l: DList(val)) → Bool
  First(l: DList(val)) → pos
  Next(p: pos, l: DList(val)) → pos
  Isend(p: pos, l: DList(val)) → Bool
  Inspect(p: pos, l: DList(val)) → val
  Insert(v: val, p: pos, l: DList(val))
    → (DList(val), pos)
  Remove(p: pos, l: DList(val)) → (DList(val), pos)
```

```

Algorithm CopyDList(l: DList)
// Return a copy of the directed list l

// Create an empty list
m ← Empty()

// Where to read in input list
p ← First(l)
// Where to insert in output list
// Will always be at the end
q ← First(m)
// For each element of the input list
while not Isend(p, l) do
    // Read value
    v ← Inspect(p, l)
    // Insert value at the end of the output list
    (m, q) ← Insert(q, v, m)
    // Advance position to be last in output list
    q ← Next(q, m)
    // Advance in input list
    p ← Next(p, l)

return m

Algorithm CopyArrayOfDList(a: Array)
// Return a copy of the array a. Each element of the array
// contains a directed list.

// Create an array with the same limits as a
c ← Create(Low(a), High(a))

// For each index in a
for i from Low(a) to High(a) do
    // Extract the list at i
    l ← Inspect-value(a, i)
    // Create a list copy
    m ← CopyDList(l)
    // Set at the proper position in the output array
    c ← Set-value(c, i, m)

// Return the array copy
return c

```


7.1 Ordning

Vilka av följande datatyper har följande egenskaper?

	Oordnad	Linjärt ordnad	Hierarkiskt ordnad	Inget av de övriga alternativen
Graf	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Hög	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> ✓	<input type="radio"/>
Binärt sökträd	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> ✓	<input type="radio"/>
Lista	<input type="radio"/>	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>
Stack	<input type="radio"/>	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>
Heltal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> ✓

Totalpoäng: 6

7.2 Sortering

Vilka av följande datatyper har följande egenskaper?

	Sorterad	Partiellt sorterad	Inget av de övriga alternativen
Graf	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> ✓
Hög	<input type="radio"/>	<input type="radio"/> ✓	<input type="radio"/>
Binärt sökträd	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>
Lista	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> ✓
Stack	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> ✓
Heltal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> ✓

Totalpoäng: 6

8.1 Komplexitet för algoritm

Betrakta algoritmen för linjärsökning till vänster. Låt varje funktionsanrop ha kostnader enligt tabellen nedan. Övriga operationer är gratis.

Operation	Kostnad
First()	1
Isend()	2
Value-isequal()	1
Inspect()	3
Next()	1
return från Linsearch	2

Låt $T(n)$ vara funktionen som beskriver **värstafallskomplexiteten** för algoritmen som en funktion av antalet element n i listan.

a) Vilken av följande former får funktionen $T(n)$?

Välj ett alternativ

☐ $an + b$



☐ $a \log n + b$

☐ $a \log(n + b)$

☐ $a^n + b$

☐ a^{n+b}

b) Ange värdena på koefficienterna a och b : $a = \boxed{} (7)$, $b = \boxed{} (5)$.

c) Vilken sorts komplexitet beskriver ovanstående fråga?

☐ Relativ komplexitet

☐ Absolut komplexitet



Totalpoäng: 8

```
Algorithm Linsearch(l: List, v: Value, Value-isequal: Function)
// Input: An unsorted list, a search value, and a equality
//        function. The Value-isequal function should accept
//        two element values and return True if the values
//        are considered equal.
// Output: (True, pos), where pos is the position for the first
//        match, or (False, None) if no match is found.

p <- First(l)
while not Isend(p, l) do
    if Value-isequal(v, Inspect(p, l)) then
        return (True, p)
    p <- Next(p, l)
return (False, None)
```

8.2 Ordo

I uppgiften nedan, om något värde behöver avrundas, avrunda till heltal på lämpligt sätt. Om flera svar är möjliga krävs optimalt värde för fulla poäng.

Antag att en algoritm har komplexiteten $T(n) = 6n + 4$.

a) Är algoritmen $\mathcal{O}(n)$ enligt ordo-definitionen?

Välj ett alternativ

☐ Ja



☐ Nej

Om svaret är Ja, ange konstanterna $c = \boxed{} (7)$ och $n_0 = \boxed{} (4 - 5)$.

b) Är algoritmen $\mathcal{O}(n^2)$ enligt ordo-definitionen?

Välj ett alternativ

☐ Ja



☐ Nej

Om svaret är Ja, ange konstanterna $c = \boxed{} (1)$ och $n_0 = \boxed{} (7)$.

Totalpoäng: 6