

## Grundläggande Datalogi för F

Tentamen 2004-01-09 kl 14.00 – 17.00

**Inga hjälpmedel.**

Endast ett svarsalternativ på varje fråga är korrekt. Felaktigt svar eller felaktigt antal ikryssade svarsalternativ ger noll poäng på uppgiften.

## Preliminära betygsgränser:

U: 0–29 poäng  
 3: 30–37 poäng  
 4: 38–44 poäng  
 5: 45–50 poäng

**Uppgift 1**

En abstrakt stack har bland annat följande egenskap.

- A) Den är oförändrad efter satsen `push(pop())`.
- B) Den är oförändrad efter satserna `push(17); pop()`.
- C) Den har en `top`-pekare.
- D) Den är en form av länkad lista.

**Uppgift 2**

En abstrakt kö har bland annat följande egenskap.

- A) Om först 1 och sedan 2 stoppas in i kön är det omöjligt att komma åt 2 utan att först ta ut 1 ur kön.
- B) Det som sist stoppats in är det som först kommer ut.
- C) Det finns en pekare till var ände av kön.
- D) När något läggs in skapas en ny nod sist i kön.

**Uppgift 3**

Antag att vi vill lagra 666 namn i en hashtabell av längden 1000 där kollisionerna hanteras med enkellänkade listor. Hur många jämförelser behövs det i medeltal för att hitta ett givet namn?

- A) Drygt en.
- B) Knappt 10.
- C) Cirka 333.
- D) Cirka 500.

**Uppgift 4**

Hashtabeller utan krocklistor har en viss dålig egenskap.

- A) Mera minneskrävande.
- B) Längre söktid.
- C) Mer komplicerad programmering.
- D) Borttagning tillåts inte.

**Uppgift 5**

Vilken av nedan föreslagna hashfunktioner är bäst? Du kan anta att nyckeln `namn` är ett efternamn. Parametern  $n$  är hashtabellens storlek, ett primtal i intervallet  $1000 < n < 100\,000$ . Pseudokoden bör vara lätt att förstå. Med `ord(tkn)` menas bokstavens ordningsnummer i alfabetet.

- A) 

```
sum = 0
for tkn in namn:
    sum = sum + ord(tkn)
return sum % n
```
- B) 

```
sum = 0
for tkn in namn:
    sum = (sum * 100 + ord(tkn))
return sum % n
```
- C) 

```
sum = 0
for tkn in namn:
    sum = sum * ord(tkn)
return sum % n
```
- D) 

```
sum = 0
for tkn in namn:
    sum = sum*17 + ord(tkn)
return sum % n
```

## Uppgift 6

En datalog som lagrar ett medlemsregister i ett binärt sökträd i stället för i en hashtabell har förmodligen följande skäl.

- A) Lättare att ta bort poster.
- B) Snabbare sökning för poster nära roten.
- C) Lättare att lagra på fil.
- D) Enklare att skriva ut i bokstavsordning.

## Uppgift 7

Sökning i ett binärt sökträd eller i en ordnad vektor, vilket går fortast?

- A) Sökträd.
- B) Vektor.
- C) Lika fort om trädet är balanserat.
- D) Lika fort om trädet är helt obalanserat (en tarm).

## Uppgift 8

För att hitta de hundra största av tiotusen värden är följande metod bäst.

- A) Sortera med quicksort, ta ut dom hundra första.
- B) Insättningssortera, avbryt när dom hundra första inte ändras.
- C) Lägg in alla i en trappa (heap) med största överst, ta sedan ut hundra.
- D) Lägg in hundra i en trappa med minsta överst, gå igenom resten och byt ibland ut trappans topptal.

## Uppgift 9

Antag att vi har en liten *hashtabell* med plats för 13 element. Antag vidare att *linjär probning* (med steget ett) används för att hantera kollisioner. Vad blir slutresultatet när strängarna "Sverige", "Norge", "Danmark", "Finland" och "Island" i denna ordning succesivt läggs in i tabellen. Vår hash-funktion,  $h$ , ger följande svar:

$h(\text{"Sverige"}) \rightarrow 3$   
 $h(\text{"Finland"}) \rightarrow 28$   
 $h(\text{"Danmark"}) \rightarrow 15$   
 $h(\text{"Norge"}) \rightarrow 25$   
 $h(\text{"Island"}) \rightarrow 12$

- A) 0: Island  
2: Finland  
3: Sverige  
4: Danmark  
12: Norge
- B) 0: Island  
2: Danmark  
3: Finland  
4: Sverige  
12: Norge
- C) 2: Finland, Danmark  
3: Sverige  
12: Norge, Island
- D) 2: Finland  
3: Sverige  
4: Danmark  
12: Island  
13: Norge

## Uppgift 10

Vilken av följande kodningar är en korrekt Huffman-kodning för data med dessa förekomstsekvenser:

'a' 35%  
'b' 21%  
'c' 15%  
'd' 13%  
'e' 11%  
'f' 5%

- A) 'a':0 'b':1 'c':01 'd':10 'e':101 'f':010
- B) 'a':10 'b':00 'c':11 'd':101 'e':100 'f':01
- C) 'a':00 'b':10 'c':010 'd':011 'e':110 'f':111
- D) 'a':0 'b':1 'c':01 'd':10 'e':10 'f':01

## Uppgift 11

Syntaxen för en binär siffra är så här.

$\langle \text{bit} \rangle ::= 0 \mid 1$

Hur är syntaxen för ett binärt heltal?

- A)  $\langle \text{tal} \rangle ::= 0 \mid 1 \mid -1 \mid \langle \text{tal} \rangle \langle \text{bit} \rangle$
- B)  $\langle \text{pos} \rangle ::= 1 \mid \langle \text{pos} \rangle \langle \text{bit} \rangle$   
 $\langle \text{tal} \rangle ::= 0 \mid \langle \text{pos} \rangle \mid -\langle \text{pos} \rangle$
- C)  $\langle \text{bitföljd} \rangle ::= \langle \text{bit} \rangle \mid \langle \text{bitföljd} \rangle \langle \text{bit} \rangle$   
 $\langle \text{tal} \rangle ::= 0 \mid 1 \langle \text{bitföljd} \rangle \mid -1 \langle \text{bitföljd} \rangle$
- D)  $\langle \text{tecken} \rangle ::= - \mid$   
 $\langle \text{tal} \rangle ::= \langle \text{tecken} \rangle \langle \text{bit} \rangle \mid \langle \text{tal} \rangle \langle \text{bit} \rangle$

## Uppgift 12

Boyer-Moore's textsökningsalgoritm är särskilt lämplig om söksträngen är

- A) lång och består av ovanliga bokstäver.
- B) kort och består av ovanliga bokstäver.
- C) lång och består av vanliga bokstäver.
- D) kort och består av vanliga bokstäver.

## Uppgift 13

Viggos rättstavningsprogram Stava har ingen ordlista, utan

- A) en boolesk hashtabell.
- B) fjorton booleska hashtabeller.
- C) ett bloomfilterträd.
- D) fjorton bloomfilterträd.

## Uppgift 14

När man tar ut översta posten ur en trappa (heap) uppstår ett tomrum. Vilken post ska man fylla det med?

- A) Bästa barnet.
- B) Sista posten.
- C) Vänstra barnet.
- D) Bästa posten på sista nivån.

## Uppgift 15

Vilken algoritm är inte exempel på principen *divide-and-conquer*?

- A) Quicksort.
- B) Mergesort:
- C) Binärsökning.
- D) Breddenförstsökning.

## Uppgift 16

Vilken sorteringsmetod kan man relativt enkelt modifiera så att den snabbt räknar ut *medianvärdet* av ett stort antal tal?

- A) Heapsort
- B) Bubblesort
- C) Quicksort
- D) Mergesort

## Uppgift 17

Stockholmsbörsen har en aktieförteckning ordnad efter aktiernas aktuella köpkurs. Det är en vektor med den dyraste aktien först. Efter varje kursändring måste programmet kolla om omsortering behövs. Vilken sorteringsmetod är lämpligast?

- A) Quicksort
- B) Bubblesort
- C) Heapsort
- D) Mergesort

## Uppgift 18

Vilken ort i världen tar det längst tid att nå från Stockholm, om man ger sig av klockan 17.00 idag? Ett program för att ta reda på svaret använder lämpligen

- A) rekursiv djupetförstsökning.
- B) breddenförstsökning med kö.
- C) djupetförstsökning med stack.
- D) bästaförstsökning med prioritetskö.

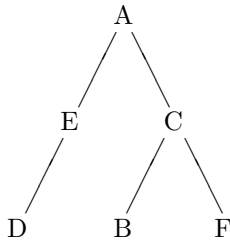
## Uppgift 19

Ett allmänt träd representeras lämpligen med följande struktur.

- A) Ett binärt träd.
- B) En länkad struktur med många pekare i varje post.
- C) En kö av stackar.
- D) En stack av köer.

## Uppgift 20

I vilken ordning behandlar man noderna vid en *postorder-genomgång* av följande träd?



- A) D, B, E, A, C, F
- B) A, B, D, E, C, F
- C) D, E, F, B, C, A
- D) D, E, B, F, C, A

## Uppgift 21

Vad menas med att ett binärt träd är *balanserat*?

- A) Alla noder i vänster delträd är mindre än roten och alla i höger delträd är större.
- B) Summan av nyckelvärdena i höger och vänster delträd är lika.
- C) Alla vägar från roten till löven är ungefär lika långa.
- D) Alla noder har antingen två eller inga barn.

## Uppgift 22

Antag att man lagrar en heap i form av en vektor så att noden vid index  $i$  har vänster barn vid  $2i$  och höger barn vid  $2i + 1$ .

Heapen har från början följande innehåll:

5	13	7	28	15	11	35	30	45	17	90
---	----	---	----	----	----	----	----	----	----	----

(första elementet har index 1)

Vad blir resultatet ifall man tar bort rotelementet och återställer heapegenskaperna?

- A) 

7	13	11	28	15	90	35	30	45	17
---	----	----	----	----	----	----	----	----	----
- B) 

7	13	17	28	15	11	35	30	90	45
---	----	----	----	----	----	----	----	----	----
- C) 

7	13	17	28	15	11	35	30	45	90
---	----	----	----	----	----	----	----	----	----
- D) 

7	13	11	28	15	17	35	30	45	90
---	----	----	----	----	----	----	----	----	----

## Uppgift 23

Att vända upp och ner på en abstrakt stack är svårt, men det går med en av följande metoder.

- A) En extra kö.
- B) En extra stack.
- C) En rekursiv metod.
- D) En extra pekare i varje stackpost.

## Uppgift 24

Vad är körtiden (*värsta fallet*) för denna algoritm?

```
algoritm värdebredd(v):  
  In: vektor v med längden n  
  
  bredd = abs(v[0]-v[1])  
  for i=0..n-2:  
    for j=i+1..n-1:  
      if abs(v[i]-v[j]) > bredd:  
        bredd = abs(v[i]-v[j])  
  return bredd
```

- A)  $\mathcal{O}(\log n)$
- B)  $\mathcal{O}(n)$
- C)  $\mathcal{O}(n \log n)$
- D)  $\mathcal{O}(n^2)$

## Uppgift 25

Vilken datastruktur använder Google för att hitta alla webbsidor som innehåller angivna sökord?

- A) En hashtabell för varje lagrad webbsida.
- B) En hashtabell för varje förekommande ord.
- C) En hashtabell med alla lagrade webbsidor.
- D) En hashtabell med alla förekommande ord.

## Uppgift 26

Det vetenskapsteoretiska stödet för att solen kommer att gå upp även i morgon, vad kallas det?

- A) Abstraktion.
- B) Ockhams rakkniv.
- C) Induktion.
- D) Falsifierbarhet.

## Uppgift 27

För att testa alla fyrsiffriga portkoder med bara 10003 tryckningar krävs en datorberäknad sifferföljd. Hur gör man bäst beräkningen?

- A) Rekursiv djupetförstsökning.
- B) Djupetförstsökning med stack.
- C) Breddenförstsökning.
- D) Bästaforstsökning.

## Uppgift 28

Maximala komprimeringen bestäms av en viss egen-  
skap.

- A) Frekvensfunktionen.
- B) Entropin.
- C) Huffmanträdet djup.
- D) Antalet löv i Huffmanträdet.

## Uppgift 29

Komprimeringsalgoritmen i Winzip är följande.

- A) RLE
- B) LZ
- C) Huffman
- D) Boyer-Moore

## Uppgift 30

En ADT specificerar följande.

- A) En datastruktur.
- B) En modul som kan importeras..
- C) En eller flera operationer.
- D) En abstrakt klass.

## Uppgift 31

Reguljära uttryck kan bland annat användas i följande situation.

- A) Vid rekursiv beräkning av fibonnaccital.
- B) När ett sårfall (exception) har uppstått.
- C) För att beskriva en ADT.
- D) Söka ord med stavningsvarianter.

## Uppgift 32

Antag att vi representerar en *stack* i form av en vektor (array) samt ett index som håller reda på stackens topp. Vektorn och indexvariablerna är från början nollställda. Vad kommer vektorn att innehålla efter nedanstående operationer?

```
s = Stack()
s.push(17)
s.push(4711)
x = s.pop()
s.push(666)
s.push(7)
```

- A) 

17	4711	0	666	7	0	0	0	0	0	...
----	------	---	-----	---	---	---	---	---	---	-----
- B) 

7	666	4711	0	0	0	0	0	0	0	...
---	-----	------	---	---	---	---	---	---	---	-----
- C) 

17	4711	666	7	0	0	0	0	0	0	...
----	------	-----	---	---	---	---	---	---	---	-----
- D) 

17	666	7	0	0	0	0	0	0	0	...
----	-----	---	---	---	---	---	---	---	---	-----

## Uppgift 33

Hur lagrar man ett binärt sökträd på fil för att kunna åter skapa det?

- A) Skriv ut det i preordning.

- B) Skriv ut det i inordning.
- C) Skriv ut det i postordning.
- D) Skriv ut det breddenförst.

## Uppgift 34

En automat specificeras av följande.

- A) Tillståndsvektorn.
- B) En funktion från indata till utdata.
- C) Övergångsmatrisen.
- D) En riktad graf där kanterna är märkta med tillstånd.

## Uppgift 35

Rekursiv medåkning kan användas i följande situation.

- A) Komprimering av bild- och ljudfiler.
- B) Beskrivning av syntax med BNF-notation.
- C) Listig omvandling från decimala till binära tal.
- D) Parsning.

## Uppgift 36

Vad är *dynamisk programmering*?

- A) Objekt skapas under körningens gång.
- B) Rekursiv beräkning där delresultat sparas.
- C) Programmet kompileras bara till en plattformsoberoende bytekod.
- D) Minnet återanvänds genom *skräpsamling*.

## Uppgift 37

För att sortera om befolkningen (cirka nio miljoner) efter födelsedag (alla födda första januari först etc) skulle det krävas så här många jämförelseoperationer. (Nio miljoner är ungefär två upphöjt till tjugotre.)

- A) Cirka tvåhundra miljoner.
- B) Cirka arton miljoner.
- C) Cirka fyrtiosex miljoner.
- D) Cirka tjugo biljoner.

## Uppgift 38

Vilken metod kan användas för att sortera en abstrakt kö?

- A) Mergesort.
- B) Heapsort.
- C) Insättningsortering.
- D) Bubbelsortering.

## Uppgift 39

Vad gör denna algoritm?

```

algoritm OK(x, y):
  p = x
  while not p is None:
    if p.data == y:
      return True
    if p.data > y:
      p = p.left
    else
      p = p.right
  return False

```

- A) Kollar att nyckeln  $x$  finns i ett binärt sökträd
- B) Kollar att nyckeln  $y$  finns i ett binärt sökträd
- C) Sätter in nyckeln  $x$  i det binära sökträdet
- D) Sätter in nyckeln  $y$  i det binära sökträdet

## Uppgift 40

En skipplista är ett alternativ till binärträd med följande egenskap.

- A) Tar mindre plats.
- B) Kräver färre pekare.
- C) Enklare att spara på fil.
- D) Mindre risk för obalans.

## Uppgift 41

En bra rekursiv tanke för siffersumman av talet  $x$  kan se ut nåt så här.

- A) Sista siffran i  $x$  plus siffersumman av  $x/10$
- B) Första siffran plus andra osv ... men 0 har summan noll.
- C) Första siffran plus siffersumman av resten.
- D) Lägg siffrorna i en stack, poppa sedan och addera..

## Uppgift 42

Antag att man har ett *binärt sökträd*. Vilken av följande algoritmer skriver ut nyckelvärdena i sorterad ordning?

- A) 

```
algoritm Mysko(root)
  if not empty(root)
    Skriv(root.key)
    Mysko(root.left)
    Mysko(root.right)
    Skriv(root.key)
```
- B) 

```
algoritm Mysko(root)
  if not empty(root)
    Mysko(root.left)
    Skriv(root.key)
    Mysko(root.right)
```
- C) 

```
algoritm Mysko(root)
  if not empty(root)
    Skriv(root.key)
    Mysko(root.left)
    Mysko(root.right)
```
- D) 

```
algoritm Mysko(root)
  if not empty(root)
    Mysko(root.left)
    Mysko(root.right)
    Skriv(root.key)
```

## Uppgift 43

Om man använder *bredden-först* för att hitta ut ur en labyrinth, vad riskerar man då?

- A) Den lösning man hittar är inte optimal.
- B) Algoritmen kan ta oändlig tid.
- C) Lösningen kan besöka samma punkt flera gånger.
- D) Algoritmen är svår att genomföra i praktiken.

## Uppgift 44

Vilken sorteringsalgoritm används när man succesivt får dessa mellanresultat i den sorterade vektorn?

13	5	9	11	3	6	8
6	5	9	11	3	13	8
6	5	3	11	9	13	8
6	5	3	8	9	13	11
3	5	6	8	9	11	13

- A) Mergesort
- B) Bubblesort
- C) Heapsort
- D) Quicksort

## Uppgift 45

Antag att man lagrat 1000 element i en ordnad vektor och konstaterar att det i medeltal tar 17 ms att hitta ett element från dess nyckelvärdet. Hur lång tid kan man förvänta sig att det tar om man istället lagrar 1000 000 element?

- A) 34 ms
- B) 170 ms
- C) 289 ms
- D) 17 s

## Uppgift 46

Vilket namn dyker upp i samband med felkorrektur?

- A) Huffman
- B) Hamming
- C) Knuth
- D) Rabin

## Uppgift 47

En Knuth-Morris-Pratt-automat för söksträngen 330331 har följande nextvektor.

- A) 0 1 2 1 1 2
- B) 0 0 2 0 0 3
- C) 0 1 0 0 1 1
- D) 0 0 1 1 1 2

## Uppgift 48

Vilken sorteringsmetod kräver mest minne.

- A) Urval
- B) Mergesort
- C) Heapsort
- D) Distributionsräkning

## Uppgift 49

Vad menas när man säger att en algoritm är *girig* (Greedy)?

- A) Den gör ett lokalt bästa drag i varje steg.
- B) Den utnyttjar maximalt med resurser.
- C) Den går igenom alla möjliga fall.
- D) Den utnyttjar minimalt med resurser.

## Uppgift 50

Antag att vi har en  $n \times n$ -matris  $A$  med bara nollor och ettor som värden, där alla ettor kommer före nollorna på varje rad. Vilken av följande algoritmer hittar *snabbast* (för stora  $n$ , värsta fallet) den rad som har flest ettor och returnerar detta antal?

- A) 

```
max = 0
for i = 1..n
  j = max+1
  while j<=n and A[i,j]==1
    j = j+1
  max = j-1
return max
```
- B) 

```
for i = n..1
  for j = 1..n
    if A[j,i] == 1
      return i
return 0
```
- C) 

```
max = 0
for i = 1..n
  count = 0
  for j = 1..n
    if A[i,j] == 1
      count = count+1
  if count > max
    max = count
return max
```
- D) 

```
max = 0
for i = 1..n
  for j = 1..n
    if A[i,j] == 1
      if j > max
        max = j
return max
```