

2D1320, TENTAMEN I TILLÄMPAD DATALOGI
Onsdag den 24 oktober 2007 kl 9–13

Maxpoäng = 100. 50 poäng ger E, men den som fått 47-49 poäng kan få komplettera. Gränserna för högre betyg är 60, 70, 80, och 90 poäng. Skriv upp antal bonuspoäng från labbar respektive hental på tentaomslaget. De rättade tentorna lämnas ut vid tentaåterlämningen i D1 kl 8-10 den 31 oktober. Hjälpmedel: En algoritmbok och ditt handskrivna formelblad. Lämnas in formelbladet tillsammans med tentan.

(14p) **1.** *Japansk granne*

- a. Konstruera och rita upp en KMP-automat som söker efter den japanska grannen TOTORO. Ange även next-vektorn!
- b. Visa hur automaten fungerar om indata är TOTOTORO
- c. Kan man skapa en KMP-automat för att känna igen ord skrivna med japanska stavelsetecken? Det finns ca 50 olika stavelsetecken, t ex ett för *TO* och ett för *RO*.

(12p) **2.** *Hasha manga*

Linda har svårt att hitta i sitt gigantiska manga-register och vill införa hashning för att få snabbare sökning. Hur ska hon göra för att det ska gå lika snabbt att söka på titel (t ex Love Hina) och författare (t ex Ken Akamatsu) utan att skapa två separata hashtabeller? Rita och förklara!

Linda köper ännu mer manga och utökar hashtabellens storlek till det dubbla. Måste hon modifiera hashfunktionen? Måste all gammal manga hashas om? Svara med motivering!

(20p) **3.** *Teori*

Nedan finns fem frågor om algoritmer och datastrukturer. Varje fråga kan ge upp till fyra poäng. Motivering krävs!

- a. Kan man använda krocklistor i ett Bloomfilter?
- b. Bör ett huffmanträd vara balanserat?
- c. Är komplexiteten för binärsökning i en vektor alltid densamma som komplexiteten för sökning i ett binärt sökträd?
- d. Varför använder man inte en prioritetskö i breddenförstsökning?
- e. Är det viktigt att hålla bägge krypteringsnycklarna hemliga vid RSA-kryptering?

(18p) 4. *Bonsaiträd*

På en textfil finns följande bonsai-typer lagrade:

- a. Tänk dig att du läser in en bonsai i taget och sorterar in den i ett binärt sökträd (sorterat i bokstavsordning). Rita upp trädet!
- b. Hur blir utskriften om trädet skrivs ut i *preorder*?
- c. Skriv en rekursiv tanke (glöm inte basfallet) för att beräkna antalet noder i ett binärt träd.
- d. Rita en serie bilder som visar hur din rekursiva tanke fungerar för trädet i uppgift a.

Netsunari
Kengai
Sekijoju
Hokidachi
Moyogi
Chokkan
Ishizuke

(16p) 5. *Japansk läsordning (tategaki)*

Våra datorer läser texter radvis, från vänster till höger. Men japansk text står i kolumner uppifrån och ner, som läses från höger till vänster. Ge en algoritm för att med hjälp av *en* abstrakt kö eller stack ordna om en japansk text om m kolumner och n rader.

七 四 一
八 五 二
九 六 三

Exempel:

7 4 1

8 5 2 läses in som 7 4 1 8 5 2 9 6 3, men vi vill ha ordningen 1 2 3 4 5 6 7 8 9

9 6 3

Du kan förutsätta att alla kolumner och rader är helt fyllda, men antalet kolumner, m , behöver inte vara detsamma som antal rader, n . Till exempel kan det finnas fem rader men bara tre kolumner.

Visa sedan hur din algoritm fungerar för exemplet ovan.

Utred till sist vilken komplexitet din algoritm har!

(20p) 6. *Kanji-syntax*

Kanji (japanska ord-tecken) är många fler än våra bokstäver och därför knepiga att skriva på västerländskt tangentbord. Ett sätt att lösa problemet är att man trycker ner flera tangenter samtidigt för att ange ett kanji. Tangenterna A...Z används, och vi numrerar kanji från 1...9999.

Tangentbordet mikroprogrammeras med en enda pythonliknande sats av typen
`if A: if B: 666 else: 667 else: if C: 668`

Här står det att om man trycker ner tangenterna A och B så skrivs tecken 666, om man trycker ner A men inte B skrivs 667, om man trycker ner C men inte A skrivs 668 och trycker man på något annat sätt händer ingenting. Observera att `elif` inte används, inte heller behöver man ha indragningar.

Skriv en syntax för satser av denna typ. Använd symbolerna $\langle sats \rangle$, $\langle tangent \rangle$ och $\langle kanjinr \rangle$.

Rita en lämplig datastruktur för att lagra ovanstående sats i. Själva orden `if` och `else` ska inte lagras i strukturen.

Beskriv hur man ur strukturen kommer fram till vilket kanji som ska skrivas om man vet vilka tangenter som är nertryckta!