

2D1320, TENTAMEN I TILLÄMPAD DATALOGI

Lördagen den 25 april 1998 kl 8–13

Maxpoäng tenta+bonus = 50+5. Betygsgränser: 25 poäng ger trea, 35 ger fyra, 45 ger femma.

Resultatet anslås troligen i morgon på Nadas anslagstavla.

Hjälpmedel : En algoritmbok.

1. *Pippiautomat*

Svenska Akademien har en databas med världens alla litterära spalter och recensioner. Den används när man bedömer nobelpristagarkandidater och för att förenkla sökning är hela textmassan är konverterad till stora bokstäver.

- (6p) Konstruera en Knuthautomat som söker efter ordet PIPPI. Rita upp den med heldragna framåtpilar och prickade bakåtpilaroch ange den next-vektor som definierar automaten!

Antyd en sökmetod som skulle gå fortare och uppskatta hur många gånger snabbare den blir.

2. *Bakvända ordningar*

- (6p) Om man ska skriva ut ett binärträd på en fil brukar man välja mellan inordning, preordning eller postordning. I alla dessa kommer vänster före höger, men ibland vill man göra tvärtom. Då får man bakvänd in-, pre- och postordning.

Rita upp ett balanserat träd med talen 1 till 7 insorterade, ange i vilken ordning talen skrivs ut på filen i dom tre fallen och rita upp det träd som skapas om man bygger ett nytt från filen!

Vilken av dom tre ordningarna kan man använda om man vill ha trädet oförändrat?

3. *Nivåutskrift*

- (6p) Ibland vill man skriva ut en viss nivå i ett binärträd. Man skulle vilja ha en PROCEDURE WriteTreeLevel(n: INTEGER; root: Pek) sådan att för sjutals-trädet i förra uppgiften anropet WriteTreeLevel(3,root) skrev ut 1 3 5 7.

Ange den rekursiva tanken, i ord eller i valfritt programspråk.

4. *Domino*

En dominobricka består av två kvadrater, vardera med noll till sex prickar på. Ett roligt tidsfördriv är att ta en näve brickor och försöka lägga ut dessa i en lång orm, där angränsande kvadrater har lika många prickar.



Du ska nu skissera ett datorprogram som läser in brickorna och sedan anger en lösning om det finns någon.

- (8p) Du behöver inte skriva någon programkod, men du ska förklara algoritmen utförligt och beskriva datastrukturer, procedurer och moduluppdelning.

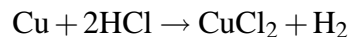
5. *Bebissortering*

Alla nyfödda vägs och mäts. Exempelvis var Elon Kann 54 cm lång och vägde 3980 g. Databasen med cirka tio miljoner poster ligger nu ordnad på personnummer, men man vill sortera om den så att långa barn kommer före korta barn. Eftersom längden bara anges i hela centimeter låter man för lika långa barn vikten avgöra – tung kommer före lätt.

(6p) Vilket sorteringsförfarande är snabbast? Uppskatta antalet jämförelser som behövs!

6. *Reaktionssyntax*

(8p) En syntax för molekylformler har förekommit i kursen. Nu gäller det att utveckla den till en syntax för reaktionsformler av typen



Skriv en grammatik som definierar symbolerna $\langle term \rangle$, $\langle summa \rangle$, $\langle reaktion \rangle$ och som använder våra tidigare kända symboler $\langle mol \rangle$ och $\langle num \rangle$ (ett heltal minst lika med 2). Beskriv kortfattat hur ditt program för molekylformelskoll kan byggas ut till att kolla att en hel reaktionsformel följer syntaxen.

Från labben är det känt hur en molekyl kan representeras med ett träd. Vilken datastruktur kan vara lämplig att representera en reaktionsformel med?

7. *Billigt bubbel*

(6p) Knuth och Sedgewick skrev var sin sorteringsprocedur för att sortera slumpmässiga dessertviner. Knuth valde en utsökt quicksort medan Sedgewick tog en billig bubble sort. När dom provkörde med tusen poster gick ändå Sedgewicks program lika fort, eftersom han har superdator. Men med tiotusen poster vann Knuth. Med hur många procent, ungefär?

8. *Abstrakta dominobrickor*

(4p) Vilken datatyp är en dominobricka? En RECORD med två fält, en tvåsiffrig INTEGER eller en ARRAY[1..2] OF INTEGER? Förklara varför ingen av dessa typer är helt idealisk. Förklara fördelen med en abstrakt datatyp istället, ange några viktiga funktioner och beskriv hur de kan implementeras i en egen modul i programspråk som Modula-3. Tänk på att datatypen ska passa för programmet i uppgift 4.