

2D1320, TENTAMEN I TILLÄMPAD DATALOGI

Lördagen den 20 mars 1999 kl 8–13

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

Resultatet anslås senast 27 mars på Nadas anslagstavla.

Hjälpmedel : En algoritmbok.

1. *Fiffig automat*

- (6p) Ett sätt att söka efter ordet FIFFIG i en textfil är att först kolla om filens bokstäver 1...6 bildar ordet, sedan kolla om filens bokstäver 2...7 bildar ordet etc. En fiffigare metod är att konstruera en Knuthautomat för ordet FIFFIG. Rita upp den med hel-dragna framåtpilar och prickade bakåtpilaroch ange den next-vektor som definierar automaten!

Uppskatta hur många gånger fiffigare den fiffiga metoden är!

2. *Nivåhöjning*

- (8p) Du ska skriva ut ett binärträd på fil, nivå för nivå, nerifrån och uppåt. För ett balanserat träd med talen 1 till 7 insorterade, skulle alltså ordningen bli 1357264. Till din hjälp har du en kö, en stack och ett gott humör.

Beskriv i ord en algoritm som löser problemet! Den ska vara så pass smart att den bara går igenom trädet en gång.

Rita också det binärträd som bildas om man bygger ett nytt från filen med 1357264.

3. *Triangelstal*

- (4p) Det n -te triangeltalet definieras som summan av dom n första heltalen. Det sjunde triangeltalet är alltså $1 + 2 + 3 + 4 + 5 + 6 + 7 = 28$.

Ge en rekursiv tanke för en funktion Triangelstal sådan att Triangelstal(7) blir 28. Ange tanken i ord eller i valfritt programspråk.

4. *Svenska Akademiens hashproblem*

- (5p) Två laborationer i kursen använder binärträd för snabb sökning i en ordlista. Ännu snabbare hade sökningen blivit med en hashvektor. Hur många gånger snabbare blir den om alla SAOLs 120259 ord ska lagras? Du kan anta att antalet jämförelser bestämmer tiden. Ange vilken storlek på vektorn du räknat med!

5. *Längst väg från fan till gud*

I en av laborationerna gällde det att gå kortaste vägen från fan till gud genom att byta ut en bokstav i taget och bara använda ordlistord. Lösningen blev

fan - ban - bad - bud - gud

(8p) Det vore intressant att veta den längsta vägen från fan till gud (utan upprepade ord).
Beskriv en algoritm för detta!

Om ordfilen innehåller cirka tusen ord, hur mycket minne kan då din metod tänkas kräva?

6. *Molekylsyntax*

Kemisterna på årets kurs har påpekat brister i molekylsyntaxen i laboration 6. En delmolekyl, alltså det som står inom parentes består alltid av minst två saker. Den kan inte vara en ensam atom eller en ensam grupp. Förbjudet är alltså $\text{Pb}(\text{H}_2)_3\text{C}_4$ medan $\text{Pb}((\text{OH})_2(\text{NO}_3)_4)_5$ är tillåtet.

(4p) Skriv en grammatik för sådana molekyler. Du kan utgå från att *<atom>* (en atombeckning) och *<num>* (ett tal större än 1) redan är definierade och kan användas i definitionen av *<mol>* (en molekyl), *<submol>* (en delmolekyl) och *<group>* (en atom eller delmolekyl, i förekommande fall med tal efter).

(3p) Beskriv i ord hur ett program kan undersöka om molekyler följer syntaxen.

7. *Lönar sej sortering?*

(8p) Etthundratusen signerade och numrerade Organzineslipsar säljs varje år. För varje såld slips sparas en post med numret och köparens namn i en array och vid årets slut lottas hundra slipsnålar av guld ut bland köparna. Det går till så att hundra vinstnummer slumpas fram, ett efter ett.

För varje nummer måste hela arrayen letas igenom, eftersom den är osorterad. Hur många jämförelser får man räkna med totalt? Lönar det sej att först sortera arrayen, en gång för alla, för att därefter sökningarna ska gå fortare?

Hade det lönat sej att spara posterna i en trappa i stället för i en array?

8. *Abstrakt blåst*

(4p) Vilken datatyp är vinden? Nordostlig storm kan representeras som en post med två textfält (`NO storm`) eller som en post med en `INTEGER` (vindriktning i grader) och en `REAL` (vindstyrka) (`45 24.0`) eller som en vindstyrkevektor med *x*- och *y*-komponenter (`17.1 17.1`) Förklara varför det är bättre med en abstrakt datatyp i stället och ange några procedurer eller funktioner för en sådan i ett program för beräkning av bästa seglingsväg!