

2D1320, Tilda, Tentamenslösning 9 mars 1996

1. *Personnummerautomat*

```
next[1..11]= 0 1 1 0 1 3 2 0 1 1 4
```

2. *Rädda regnskogen!*

Man ska använda **preordning**.

3. *Balanskonst*

```
PROCEDURE ReadTree(n: INTEGER): Pek RAISES {}=  
  VAR m:= n DIV 2;  
      p:=NEW(Pek);           Skapa en tompost för roten!  
  BEGIN  
    IF n=0 THEN RETURN NIL END;  
    p.left:=ReadTree(m);     Bygg vänster delträd av halva filen!  
    p.word:=IO.GetLine();    Läs in mittordet i rotposten!  
    p.right:=ReadTree(n-m-1); Bygg höger delträd av halva filen!  
    RETURN p;  
  END ReadTree;
```

4. *Längre vardag*

Problemträdet består av ordposter med faderspekare. Det tomma ordet är stamfar. Man gör breddenförstsökning med hjälp av en kö ända tills kön tömts. Det sista ordet ger då lösningen via faderspekarkedjan. Söner skapas genom att en bokstav läggs till först eller sist.

Alla hashfunktioner beräknas i en slinga av typen

$$h := (h * 128 + \text{ordets } j\text{:te bokstav}) \bmod \text{primita}$$

med olika primita för olika funktioner. Det är lämpligt att gå in i slingan med ett stort h-värde, annars får korta ord fjorton lika hashvärden.

5. *Diamantsortering*

Eftersom ordningen inte ska ändras så mycket är insättningsortering bäst. Men om man bara får plats med hundra poster i taget är det bäst att använda dom till en trappa. En genomkörare räcker då förutsatt att ingen post ligger mer än hundra positioner fel.

6. *Reaktionssyntax*

```
<term>      ::= <mol> | <num><mol>
<summa>     ::= <term> | <term> + <summa>
<reaktion> ::= <summa> -> <summa>
```

`ReadReaktion()` anropar `ReadSumma()`, glufsar en pil och anropar `ReadSumma()` igen, `ReadSumma()` anropar `ReadTerm()`, kollar om det står ett plus, glufsar i så fall det och anropar `sej` själv, `ReadTerm()` anropar kollar om det står en siffra och anropar i så fall `ReadNum()`, sedan anropas i vilket fall som helst `ReadMol()`.

7. *Billig standard selection*

När N växer från tusen till tiotusen växer $N^2/N \log N$ med faktorn sju ungefär, så Tildas sortering tog nog cirka en sjundedel så lång tid som Torstens.

8. *Abstrakta personnummer*

En `INTEGER` får inte vara hur stor som helst, tiosiffriga pnr kan bli för stora. En `ARRAY[1..10] OF INTEGER` tar mycket minnesutrymme (fyrtyo byte). En `TEXT` är det bökigt att plocka ut t ex årtal ur. Väljer man en abstrakt datatyp kan man sedan välja den bästa implementationen genom att bara ändra i modulen `Pnr.m3`. Ännu mera abstrakt blir det med ett objekt:

```
INTERFACE Pnr;

EXCEPTION WrongCheckdigit;
TYPE T<: OBJECT
    METHODS
        male(): BOOLEAN;
        female(): BOOLEAN;
        year(): INTEGER;
        month(): INTEGER;
        day(): INTEGER;
        yeardays(): INTEGER;
        equal(pnr:T): BOOLEAN;
        before(pnr:T): BOOLEAN;
        totext(): TEXT    ;
        fromtext(word:TEXT) RAISES WrongCheckdigit;
END Pnr.
```