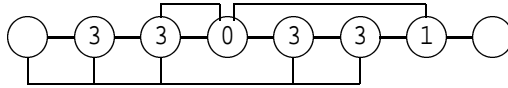


2D1320, Tilda, Tentamenslösning 21 mars 1998**1. Knuthautomat**

i	next[i]
1	0
2	0
3	2
4	0
5	0
6	3

Boyer–Moore tittar i stället på vart sjätte tecken. Så länge det inte är en trea, etta eller nolla hoppar man vidare. Man tittar bara på en sjättedel så många tecken men gör en mer komplicerad jämförelse, så det kanske tar en tredjedel så lång tid.

2. Ordna stacken efter kön

- WHILE NOT Stack.Empty() DO Queue.Put(Stack.Pop()) END;
- WHILE NOT Queue.Empty() DO Stack.Push(Queue.Get()) END;
- WHILE NOT Stack.Empty() DO Queue.Put(Stack.Pop()) END;
- Lägg pnyxtr sist i kön och gör följande upprepade gånger.
 - * Ta en post ur kön.
 - * Om det är en man pushas han på stacken.
 - * Om det är en kvinna läggs hon sist i kön.
 - * Om det är pnyxtr avbryts slingan.
- WHILE NOT Queue.Empty() DO Stack.Push(Queue.Get()) END;

3. Rekursiv spegling

```

PROCEDURE Spegla(p:Pek)=
  VAR temp:Pek;
  BEGIN
    IF p=NIL THEN RETURN END;
    temp:=p.left; p.left:=p.right; p.right:=temp;
    Spegla(p.left); Spegla(p.right);
  END Spegla;

```

4. Krockade ordlistor Den första funktionen skiljer inte *AB* från *BA* och är därför sämre. Viggos program finns på kurssidan.

5. Problem i patiensbok

Rekursiv djupetförstsökning är bäst. Man kan ha en global 4×4 -matris med utlagda kort och en global kö med oanvända kort. Först är matrisen tom och alla sexton korten finns i kön. Anropet Place(1) tar ett kort ur kön, lägger ut det längst upp till vänster, gör

anropet Place(2) som tar ett kort ur kön och lägger ut det på nästa plats etc. Proceduren som lägger ut kort i matrisen kollar samtidigt horisontellt, vertikalt och diagonalt att det är tillåtet.

När matrisen fullbordats skrivs den ut, men programmet bryts inte.

Modulerna Queue, Matrix och Card behövs. Card.i3 kan se ut så här:

```
INTERFACE Card;
TYPE T<:REFANY;
PROCEDURE New(nr:INTEGER):T;
PROCEDURE Alike(card1,card2:T):BOOLEAN;
PROCEDURE Write(card:T);
END Card.
```

6. Syntax för funktionsanrop

```
<funktionsanrop> ::= <funktion>(<parameterlista>)
<variabel>       ::= a | b | ...
<funktion>       ::= A | B | ...
<parameter>      ::= <variabel> | <funktionsanrop>
<parameterlista> ::= <parameter> | <parameter>, <parameterlista>
```

Proceduren ReadFunktionsanrop anropar ReadFunktion, kollar och glufsar vänsterparentes, anropar ReadParameterlista, kollar och glufsar högerparentes. Proceduren ReadParameterlista anropar ReadParameter och tjuvtittar på nästa tecken. Om det är ett komma glufsas det och ett rekursivt anrop följer.

7. Tio i topp

En tioposterstrappa med *minsta* värdet överst är perfekt. Hashvektorn är helt oduglig, binärträdet nästan odugligt.

8. Gyllene tider

Med en abstrakt Datum.T kan implementeringen Datum.m3 ändras utan att det påverkar något annat i programmet. Gränssnittet Datum.i3 bör bland annat innehålla anropen

```
ToText(T):TEXT; (* "21 maj 1998" *)
FromText():T; (* enl ovan eller "980321" *)
Year(T):INTEGER; (* 1998 *)
Weekday(T):TEXT; (* lördag *)
Interval(d1,d2:T):INTEGER; (* antal dagar d1 till d2 *)
```