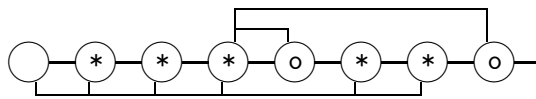


2D1320, Tilda, Tentamenslösning 16 januari 2001

1. *Tjugondag Knut kastas julen ut*

Distributionsräkning är bäst ($\sim N$) eftersom det bara finns 365 olika datum. Trappsortering är näst bäst ($\sim N \log N$) och man kan avbryta när hälften sorterats. Insättning fungerar också ($\sim N^2$). Hashning är nästan oanvändbart; man bör i så fall vara säker på att hashfunktionen inte kan ge krockar. Binärsökning och djupet-först-sökning är det bara att glömma.

2. *Tjugondagknuthautomat*

i	next[i]
1	0
2	0
3	0
4	3
5	0
6	0
7	3

Det går inte att snabba upp sökningen nämnvärt. Boyer-Moore skulle inte fungera eftersom alla tecken finns med i söksträngen. En liten förbättring vore att strunta i jämförelsen när man backat till tillstånd 3 och i stället glufsa stjärnan och gå fram till 4.

3. *Julprydnadsstack*

Preordning \Rightarrow D B A C F E G

Inordning \Rightarrow A B C D E F G

Postordning \Rightarrow A C B E G F D

Man ska använda **postordning**.

4. *Rekursiva granbarr*

Om man inte har något barn under sej är man själv ett barr och då är barrantalet lika med $(1 + \text{barrantalet till höger})$, annars är det $(\text{barrantalet nedanför} + \text{barrantalet till höger})$.

...men ett tomt träd har inga barr.

5. *Godispåskonstruktion*

Sortera smågodissorterna med den tyngsta först. Vi utgår från **godis**, en vektor av sjutton heltal, alltså antal godis av varje sort, och **vikt**, en vektor av sjutton flyttal. Totalvikten beräknas, divideras med tio och ger då **idealvikt**.

Algoritmen börjar med tio tomma påsar (stamfar i problemträdet) och börjar med att fylla första påsen godis för godis. Stamfadern har alltså sjutton söner.

För att slippa dumsöner tar man alltid tunggodis före lättgodis. När en påse kommer högst ett gram från idealvikt börjar nästa.

Djupetförst är bäst eftersom man inte är ute efter kortaste lösningen. Den kan göras rekursivt och i så fall blir den enda datastrukturen påsmatrisen med 17×10 heltal.

6. *Ringdanssyntax*

```
<upprepning> ::= <siffra> [ <ringdans> ]  
<byggsten>  ::= <ord> | <upprepning>  
<ringdans>  ::= <byggsten> | <byggsten><ringdans>
```

7. *Fiskdammsdatastrukturer*

Presenterna läggs i en kö och barnen i en trappa sorterad på värde+ålder där värdet mäts i ören.

8. *Abstrakta julgransnummer*

Antal siffror i datum osv kan komma att ändras. För att inte låsa sej i en viss implementering bör man ha en abstrakt trädtyp Träd med följande metoder.

```
int year();  
int region();  
int number();  
void setYear(int year);  
void setRegion(int region);  
void setNumber(int number);  
String toString()
```