

2D1320, Tilda, Tentamenslösning 18 mars 2000

1. *Fruktautomat*

i	1	2	3	4	5	6		Tillståndsföljd:
next[i]	0	1	0	1	0	4		10 1 2 3 4 5 64 5 64 5 6 7

2. *Fruktsortering*

Säg att $\text{Merge}(8)$ samsorterar dom första 8 frukterna i kön med dom 8 frukterna i en extrakö och lägger resultatet sist i originalkön. Vi antar att dom båda 8-följderna redan är sorterade var för sej och att när dumfrukten pnyxtr dyker upp Merge bara tömmer extrakön till originalkön och flyttar pnyxtr sist.

Lägg dumfrukten pnyxtr sist i fruktkön, sätt $k=1$ och gör så här:

- Flytta k frukter till extrakön och gör $\text{Merge}(k)$.
- Om pnyxtr dök upp är vi klara.
- Annars upprepar vi tills pnyxtr visar sej, lägger pnyxtr sist i kön igen, fördubblar k och gör om alltihop.

3. *Frukträd*

Antalet frukter på nivå k är antalet på nivå $k - 1$ i vänsterträdet plus antalet på nivå $k - 1$ i högerträdet ...men tomt träd har noll frukter ...och annars om $k = 0$ är antalet 1.

På nivå 7 finns det 128 frukter.

4. *Fruktkvalitet*

Problemträdet har ett tomt ord som stamfar, åtta enbokstavsord som söner, 56 tvåbokstavskombinationer som sonsöner etc. Rekursiv djupetförst är bäst. $\text{MakeSons}(\text{"avl"}, \text{"kitet"})$ skapar fem söner till avl genom att klistra på en bokstav ur kitet på slutet, kolla med binär sökning i SAOL-vektor om det finns något ord som börjar så och i så fall anropa vidare rekursivt. När ett helt SAOL-ord upptäcks läggs det in i ett dumsonsträd för att inte räknas dubbelt.

5. *Fruktad koll*

- Räkna förekommande bokstäver i en frekvensvektor.
- Kolla SAOL-existens med Viggos Stava (utan böjningsformer).
- Insättningssortera diskettorden och kolla sedan upprepningar.

6. *Fruktsalladssyntax*

```
<fruktsallad> ::= <frukt><mellan>SALLAD
<mellan> ::= <nothing> | - <frukter> OCH <frukt>
<frukter> ::= <nothing> | , <frukt>- <frukter>
```

Proceduren `ReadFruktsallad` anropar `ReadFrukt`, `ReadMellan` och åter upp `SALLAD`. `ReadMellan` tjuvtittar om det kommer bindestreck, i så fall glufsas det varefter `ReadFrukt` anropas, `OCH` glufsas och `ReadFrukt` anropas. `ReadFrukt` tjuvtittar om det kommer ett komma, i så fall glufsas det, `ReadFrukt` anropas, bindestreck glufsas varefter `ReadFrukt` anropas rekursivt.

7. *Fruktgallring*

Man gallrar stacken genom att poppa översta frukten, kasta den, om stacken inte är tom poppa en frukt till och spara den, gallra den stack som är kvar och pusha tillbaka den sparade frukten.

...men en tom stack behöver inte gallras.

8. *Abstrakt fruktregister*

Hashtabeller kan inte skrivas ut i bokstavsordning, binärträd har inte blixtsnabb sökning. En kombination är förmodligen bäst, där både en hashtabell och ett binärträd används. Men i stället för att det anropande programmet ska behöva göra anropen `Hash.Get("banan")` och `BinTree.WriteAll()` bör man ha ett abstrakt register `Fruktreg.i3` med bland annat följande anrop.

```
Put(name:TEXT;data:REFANY);  
Get(name:TEXT):REFANY;  
Delete(name:TEXT);  
WriteAll();
```