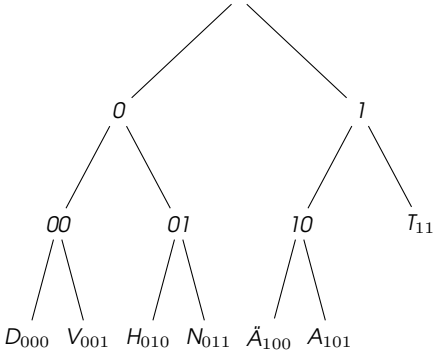
Per Sedholm
DD1320 (tilda12)
2012-10-04

Övning 6

Komprimering, kryptering, dokumentering & testning

1. Smittskydd

Du har fått ett mail som innehåller tips mot spridning av virus. Informationen är komprimerad med ett Huffmanträd där nollor motsvarar vänster och ettor motsvarar höger (se figur, "T" kodas t.ex. som `11`*) Vad står det i meddelandet* `010101011000110011001111`*?*



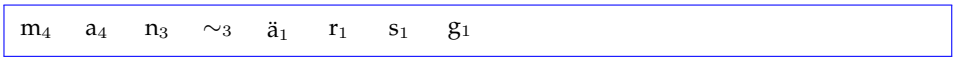

```
010 101 011 000 11 001 100 11 11
 H   A   N   D  T  V   Ä  T  T
```

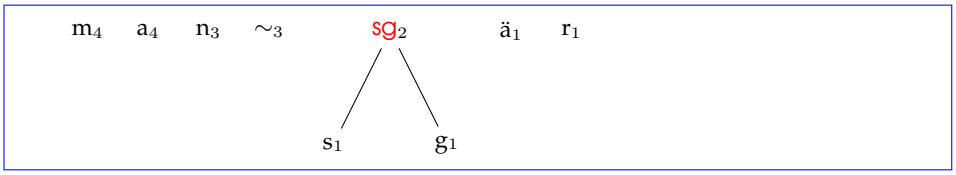
2. Huffman för Havamal

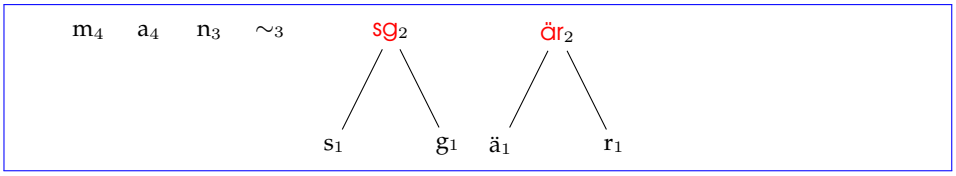
"man är mans gamman" kan man läsa i Havamal. Konstruera en huffmankod för tecknen i detta uttryck (rita trädet) och skriv sedan upp det i kodad form.

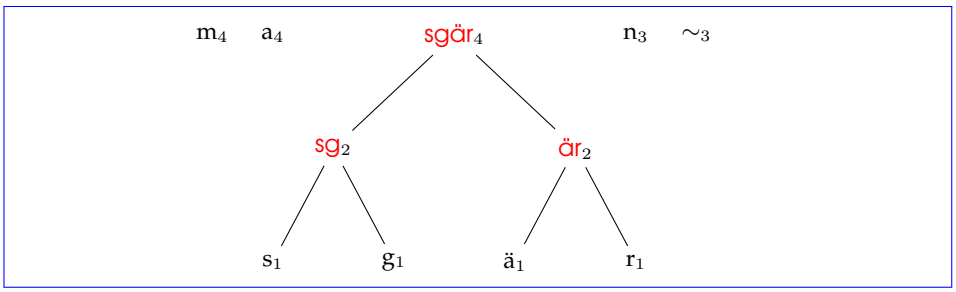
Bokstav	m	a	n	␣	ä	r	s	g
Frekvens	4	4	3	3	1	1	1	1

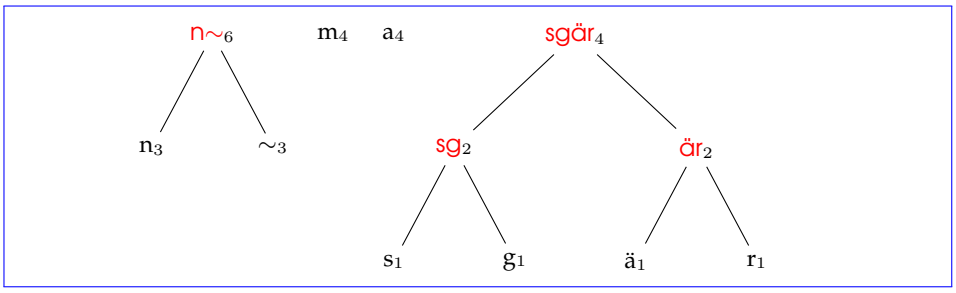
Trädet växer fram som:

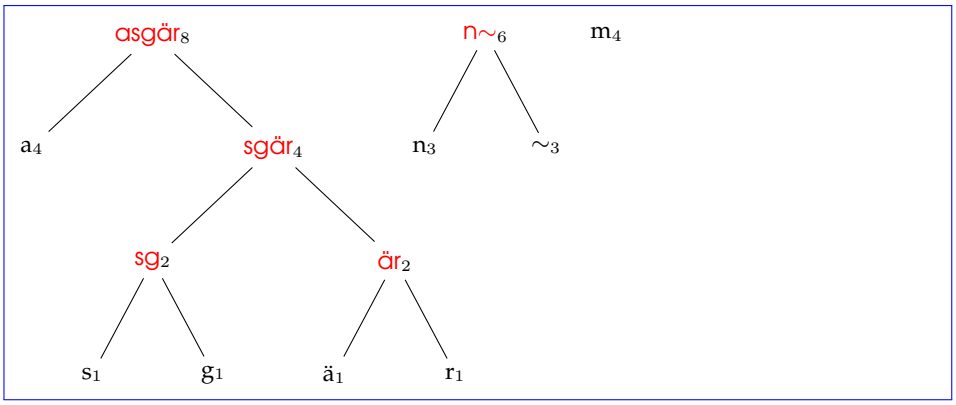


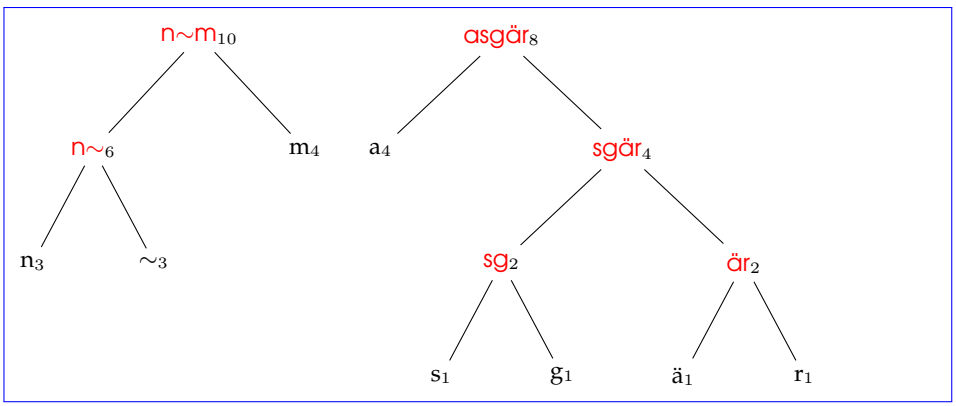


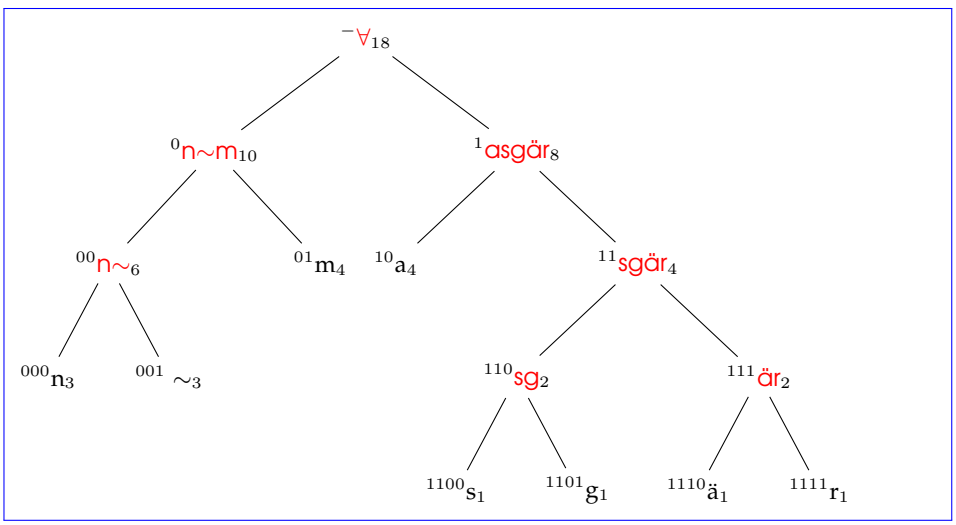



m4 a4 sgär4 n3 ∼3
sg2 är2
s1 g1 ä1 r1


n∼6 m4 a4 sgär4
n3 ∼3 sg2 är2
s1 g1 ä1 r1


asgär8 n∼6 m4
a4 sgär4 n3 ∼3
sg2 är2
s1 g1 ä1 r1


n∼m10 asgär8
n∼6 m4 a4 sgär4
n3 ∼3 sg2 är2
s1 g1 ä1 r1


¯∀18
0n∼m10 1asgär8
00n∼6 01m4 10a4 11sgär4
000n3 001 ∼3 110sg2 111är2
1100s1 1101g1 1110ä1 1111r1

Med vänster → 0, höger → 1 får vi

Bokstav	n	␣	m	a	s	g	ä	r
Kod	000	001	01	10	1100	1101	1110	1111

Koden för ”man är mans gamman” blir

```
m  a  n  _  ä  r  _  m  a  n  s  _  g  a  m  m  a  n
01 10 000 001 1110 1111 001 01 10 000 1100 001 1101 10 01 01 10 000

01100000 01111011 11001011 00001100 00111011 00101100 00
       8        16       24       32       40       48    (50 bitar)
```

En alternativ lösning (se hemsida) ger koden

Bokstav	ä	r	s	g	n	␣	m	a
Kod	0000	0001	0010	0011	010	011	01	11

```
10 11 010 011 0000 0001 011 10 11 010 0010 011 0011 11 10 10 11 010

10110100 11000000 01011101 10100010 01100111 11010110 10
       8        16       24       32       40       48    (50 bitar)
```

3. Enkel kryptering

Kryptera lösenordet `SIMSALABIM` *med*

1. *rot13*
2. *Transpositionschiffer*

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
```

1. `FVZFNYNOVZ`
2. `BISALAMIMS` (mfl)

4. Testning

I labb 6 ska ni göra ett program som kontrollerar syntaxen för en molekylformel. Skriv upp en samling indata att testa programmet med!

- Giltiga formler
- Ogiltiga formler
- Tomma strängen?
- Både en- och tvåteckens siffror
- Tänk på att ej korrekta kemiska formler är tillåtna.

5. RSA-kryptering (ur kursboken, uppg 2 s. 125)

Du vill hålla ditt turnummer (18) hemligt, och har därför bestämt dig för att kryptera det med RSA. Välj två primtal större än 500 och använd dom för att kryptera turnumret.

$$\left.\begin{array}{rcl} p & = & 587 \\ q & = & 719 \end{array}\right\} n = p \cdot q = 422\,053$$

$$\phi(n) = (p-1) \cdot (q-1) = 420\,748 = 2 \cdot 2 \cdot 293 \cdot 359$$

$e = 17$ väljs så att $1 < e < \phi$, och e och ϕ är *relativt prima*, dvs inga faktorer gemensamma, eller $gcd(e, \phi) = 1$.

Kryptering P av meddelandet M fås genom*

$$\begin{array}{rcl} P(M) & = & M^e \pmod{n} \\ P(18) & = & 18^{17} \pmod{422\,053} \\ & = & 2.185\ldots \cdot 10^{21} \pmod{422\,053} \\ & \equiv & 161\,344 \pmod{422\,053} \end{array}$$

För att avkryptera chiffertexten C används

$$S(C) = C^{\,d} \pmod{n}$$

där $d = 222\,749$ beräknas som modulär multiplikativ invers (*modular multiplicative inverse*, se nedan), dvs d sådant att

$$d \equiv e^{-1} \pmod{\phi(n)}$$

Här kan se att

$$\begin{array}{rcl} S(161\,344) & = & 161\,344^{222\,749} \pmod{422\,053} \\ & = & 5.405\ldots \cdot 10^{1\,160\,021} \pmod{422\,053} \\ & \equiv & 18 \pmod{422\,053} \end{array}$$

Det fungerar även ”baklänges”. För att signera meddelandet $C^* = 18$:

$$\begin{array}{rcl} S(C^*) & = & 18^d \pmod{n} \\ & \equiv & 197\,036 \pmod{n} \\ P(197\,036) & = & 197\,036^e \pmod{n} \\ & \equiv & 18 \pmod{n} \end{array}$$

Paret (e, n) är din allmänna nyckel (*public key*).

Paret (d, n) är din privata nyckel (*private key*).

För att signera, beräkna med privata nyckeln så kan andra verifiera med den allmänna. För att kryptera kan andra beräkna med den allmänna, och du avkryptera med den privata.

*Vid behov fyller man ut M till önskad bitlängd

Modulär multiplikativ invers

Antag $a = 3$. Då är $b = 7$ dess invers modulo 10:

$$a^{-1} \equiv b \pmod{10} \longleftrightarrow a \cdot b = 3 \cdot 7 = 21 \equiv 1 \pmod{10}$$

Istället för att dividera med a kan man då multiplicera med b:

$$\frac{n}{a} = n \cdot a^{-1} = n \cdot b$$

Exempel:

$$\begin{array}{llll} n = 12 & 12/3 = 4 \equiv & 4 & \equiv 84 = 7 \cdot 12 \pmod{10} \\ n = 15 & 15/3 = 5 \equiv & 5 & \equiv 35 = 5 \cdot 7 \pmod{10} \\ n = 588 & 588/3 = 196 \equiv & 6 & \equiv 4116 = 5 \cdot 7 \pmod{10} \end{array}$$

För att en invers till a ska existera i $\mathbb{Z}_m$ så måste a och m vara relativt prima.

Inversen beräknas med en utökad version av Euklides algoritm. I RSA-exemplet ovan är

$$d \cdot e = 17 \cdot 222\,749 = 3\,786\,733 \equiv 1 \pmod{420\,748}$$

6. Lempel-Ziv

I denna uppgift ska du avkoda ett meddelande som komprimerats med Lempel-Zivs metod. Komprimeringen går till så att komprimeraren lagrar en lista med strängar som från början enbart innehåller tomma strängen.

Komprimeraren läser tecken för tecken från intexten den längsta sträng som ligger i strängtabellen. Sedan skrivs index för denna sträng i tabellen ut, följt av nästa tecken i intexten.

Strängtabellen utökas med strängen plus nästa tecken. Därefter läses åter tecken för tecken från intexten.

Så fort en sträng inte finns i strängtabellen läggs den alltså till.

Metoden kan i (icke-optimerad) kod beskrivas så här:

```
def lzw(text):
    table = Table()
    q=Queue()
    for c in text: # spara texten tecken för tecken i en kö
        q.put(c)
    s=""
    table.add(s)
    kodtext="" # här sparas den kodade texten
    while not q.isempty():
        c=q.get()
        if table.exists(s+c):
            s=s+c
        else:
            kodtext+=str(table.code(s))+c
            table.add(s+c)
            s=""
    if not s=="":
        kodtext+=str(table.code(s))
    return kodtext
```

Klassen Table *stöder inläggning av strängar, kontroll av om en sträng finns i tabellen och möjlighet att ta reda på index hos en speciell sträng.*

Index bestäms av ordningen strängarna lades in i, där den första strängen har index 1.

Uppgiften är att avkoda följande text, där alfabetet består av versaler och mellanslag kodat som ␣.

`1S1T1O1R2T4C1K1H4L1M2_6O1L3A1_9O14M1A5`

Det vill säga: vad är t *om* `print lzw(t)` *ger ovanstående som utskrift?*

Ge också en tabell med strängar och index som motsvarar tabellen i koden!

index	1	2	3	4	5	6	7	8	9	10	11	12
sträng	””	S	T	O	R	ST	OC	K	H	OL	M	S␣

index	13	14	15	16	17	18	19
sträng	STO	L	TA	␣	HO	LM	A

`1S1T1O1R2T4C1K1H4L1M2_6O1L3A1_9O14M1A5`

”S T O R ST OC K H OL M S STO L TA ␣ HO LM A R”