1. Übung aus Informations- und Codierungstheorie
SS 2016

1. Sei $P = (p_1, \ldots, p_m)$ eine Wahrscheinlichkeitsverteilung (o.E.d.A.: $p_1 \geq \ldots \geq p_m$). Mit der Rekursionsformel:

$$H^*(P) = 1 + \min_{A \subset \{1,\ldots,m\}} (P(A)H^*(P(\, . \, |A) + P(A^c)H^*(P(\, . \, |A^c))$$

berechne man $H^*(P)$ für $m = 2, \ldots, 5$.

Lösung :

$$\begin{aligned}
m = 1: \quad & H^*(P) = 0 \\
m = 2: \quad & H^*(P) = 1 \\
m = 3: \quad & H^*(P) = 1 + \min_{1 \leq i \leq 3} \{0 \cdot p_i + (1 - p_i) \cdot 1\} = 2 - p_1 \\
m = 4: \quad & |A| = 1 : H^*(P) = 1 + \min \left\{ 0 \cdot p_i + (1 - p_i) \left(2 - \max_{j \neq i} \tfrac{p_j}{1-p_i} \right) \right\} = \\
& = 3 + \min_i \{-2p_i - \max_{j \neq i} p_i\} = 3 - 2p_1 - p_2 \\
& |A| = 2 : H^*(P) = 1 + \min_{i_1, i_2} \{(p_{i_1} + p_{i_2}) \cdot 1 + (1 - p_{i_1} - p_{i_2}) \cdot 1\} = 2 \\
& \Longrightarrow H^*(P) = \min\{2, 3 - 2p_1 - p_2\} \\
m = 5: \quad & |A| = 1 : \\
& 1 + \min_i \{0 \cdot p_i + (1 - p_i) \cdot \min\{2, 3 - 2 \cdot \max_{j \neq i} \tfrac{p_j}{1-p_i} - \max_{(2) j \neq i} \tfrac{p_j}{1-p_i}\} = \\
& = 1 + \min_i \{\min\{2 - 2p_i, 3 - 3p_i - 2 \cdot \max_{j \neq i} p_j - \max_{k \neq i,j} p_k\}\} = \\
& = 1 + \min\{2 - 2p_1, 3 - 3p_1 - 2p_2 - p_3\} = \\
& = \min\{3 - 2p_1, 4 - 3p_1 - 2p_2 - p_3\} \\
& |A| = 2 : 1 + \min_{i,j} \left\{ (p_i + p_j) \cdot 1 + (1 - p_i - p_j) \left(2 - \max_{k \neq i,j} \tfrac{p_k}{1-p_i-p_j} \right) \right\} = \\
& = 1 + \min_{i,j} \{1 + 1 - p_i - p_j - \max_{k \neq i,j} p_k\} = \\
& = 3 - p_1 - p_2 - p_3 \\
\text{Somit} \quad & H^*(P) = \min\{3 - 2p_1, 4 - 3p_1 - 2p_2 - p_3, 3 - p_1 - p_2 - p_3\}
\end{aligned}$$

2.
```
Huffman <- function(p,x=c(1:length(p)),D=2) {
    # Die Funktion berechnet die Codeworte fuer eine Quelle, die die Ausgaenge
    # x mit W-keit p sendet. D ist die Groesse des Codealphabets.
    # Eingabe: p (kann auch ein Vektor von Haeufigkeiten sein)
    # x Vorgabe 1:length(p)
    # D Vorgabe 2
    # Ausgabe: x , die Codeworte
    # C(x) (die Buchstaben des Codealphabets werdenals durch Komma getrennte
    # Zahlen dargestellt, um D nicht auf 10 oder 26 beschraenken zu muessen).
    # m2 die Nummer der Wurzel
    # V die Matrix der Vorgaenger. Der Codebaum wird gezeichnet, indem von der
    # Wurzel Kanten zu ihren Vorgaengern gezogen werden. Knoten mit Nummern <=
    #m2:= m+(D-m)mod (D-1) stellen Blaetter dar, die keine Vorganger mehr haben.
    # Von Knoten mit Nummern > m2 werden wieder Kanten zu ihren Vorgangern
    # gezogen usw.
    m <- length(p);             # die Anzahl der Ausgaenge
    m2 <- m1 <- m+(D-m)%%(D-1); #fuege Ausgaenge hinzu,sodass m2=0mod(D-1)
    H <- matrix(0,3,m2);        # Matrix mit W-keiten, Ausgaengen u.
    H[1,1:m] <- p;             # Nummern f. die Ausgaenge
    H[2,1:m2] <- 1:m2;
    H[3,1:m] <- x;
    V <- matrix(0,m2,D);        # initialisiere Matrix der Vorgaenger
    H <- H[,order(H[1,],decreasing=T)]; # ordne H nach fallenden W-keiten
    G <- H;                       # Hilfsmatrix
    i <- 0;                       # Laufindex
    while (m1 > 1)
      { i <- i+1;
          G[1,m1-D+1]<-sum(G[1,(m1-D+1):m1]) ;
           h<- c(G[1,m1-D+1],m2+i,m2+i);
                    V[i,1:D] <- G[2,m1-D+c(1:D)];
                    H <- cbind(H,h);
                    G <- cbind(G[,1:(m1-D)],h);
                    G <- G[,order(G[1,],decreasing=T)];
                    m1 <- m1-D+1 };
    C1 <- 0;
    C1[V[i, ]] <- as.character( c(0:(D-1)) ) ;
    for (j in 1:(i-1)){
      C1[ V[i-j,]] <- rep(C1[m2+i-j],D) ;
     for ( k in 1:D){ C1[V[i-j,k]] <- paste(C1[ V[i-j,k ]],",",k-1) };
    };
         V1 <- cbind(c((m2+i):(m2+1)), V[i:1,]);
         colnames(V1) <- c("Nachfolger", rep("Vorgaenger",D));
         rownames(V1) <- rep(" ",i);
        list(Ausgaenge=x[1:m],Codeworte=C1[1:m],Wurzelnummer=m2+i,Vorgaenger=V1 )
```