

姓名: 劉哲安

學號: 410886041

系級: 通訊系 三年級

一. huffman code

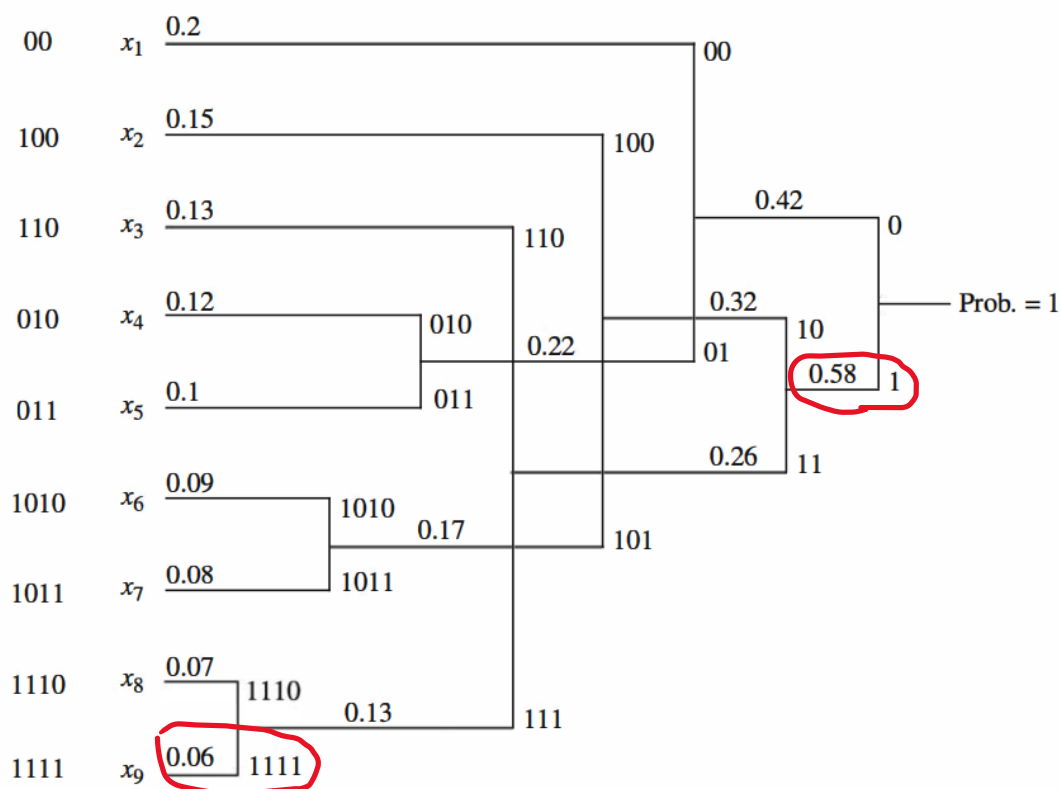
1. 編碼方式(目的):

出現機率高的字母使用較短的編碼(位元使用較少)，反之出現機率低的則使用較長的編碼(位元使用較多)，這便使編碼之後的字串的平均長度、期望值降低，從而達到無失真壓縮數據的目的。

以參考書上的圖為例子可以看到

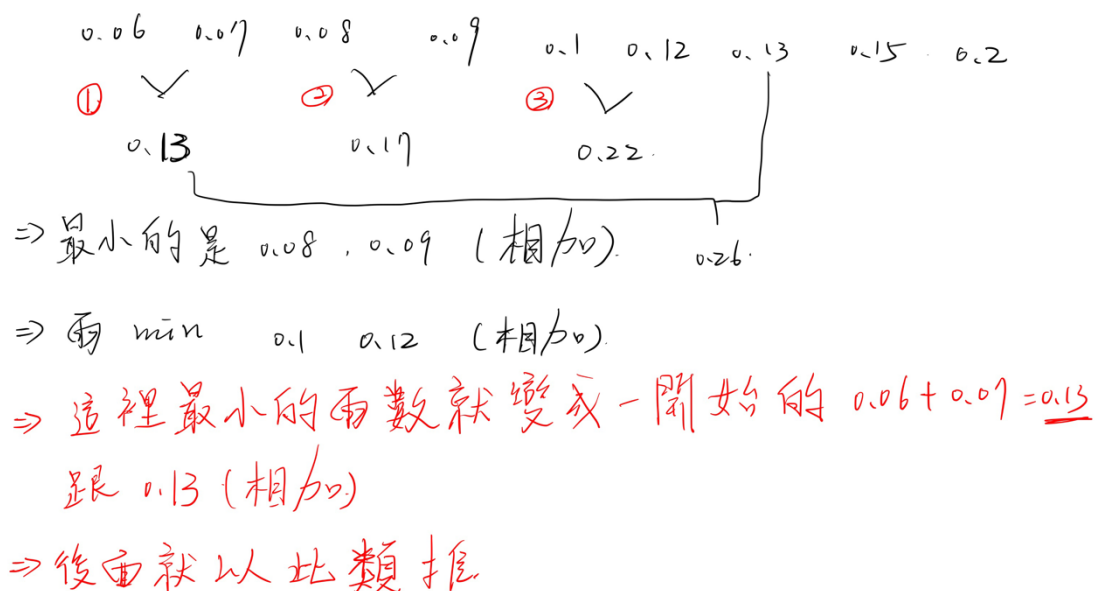
0.06(出現最少)用 1111 編碼(最長),而 0.58(出現最多) 用 1 編碼(最短)並且機率總合為 1

Codewords



第一步:建構 Huffman tree 前要先將數由小到大排列

第二步:每次都挑最小的兩個節點合併



2.matlab code(Huffman function)

```
function [h,l]=huffman(p);
%HUFFMAN Huffman code generator.
% [h,l]=huffman(p), Huffman code generator
% returns h the Huffman code matrix, and l the
% average codeword length for a source with
% probability vector p.
if length(find(p<0))~=0,%機率小於 0(錯誤)
    error('Not a prob. vector, negative component(s)')
end
if abs(sum(p)-1)>10e-10,%機率之和大於 1(錯誤)
    error('Not a prob. vector, components do not add up to 1')
end
n=length(p);
q=p;
m=zeros(n-1,n);
for i=1:n-1
    [q,l]=sort(q);
    m(i,:)=[l(1:n-i+1),zeros(1,i-1)];
    %由 l 建構一個矩陣,表明合併時的順序,用於後面的編碼
    q=[q(1)+q(2),q(3:n),1];
end
for i=1:n-1
    c(i,:)=blanks(n*n);%矩陣進行 Huffman 編碼
end
c(n-1,n)='0';
c(n-1,2*n)='1';
%由於矩陣的第 N-1 列的前兩個元素為進行編碼的總和運算的最後兩個值,0 or 1
%大的為 1,小的為 0
for i=2:n-1
    c(n-i,1:n-1)=c(n-i+1,n*(find(m(n-i+1,:)==1))...
        -(n-2):n*(find(m(n-i+1,:)==1)));
    % 矩陣 c 的第 n-i 的第一個元素的 n-1 的字符的值對應於矩陣中第 n-i+1 列中值為 1 的位
    % 置在 c 矩陣中的編碼值
    c(n-i,n)='0';
    c(n-i,n+1:2*n-1)=c(n-i,1:n-1);
```

%矩陣 c 的第 n-i 的第二個元素的 n-1 的字符與第 n-i 列的第一個元素的前 n-1 個符号相同，因為其根節點相同

```
c(n-i,2*n)='1';
for j=1:i-1
    c(n-i,(j+1)*n+1:(j+2)*n)=c(n-i+1,...
    n*(find(m(n-i+1,:)==j+1)-1)+1:n*find(m(n-i+1,:)==j+1));
end
end
for i=1:n
    h(i,1:n)=c(1,n*(find(m(1,:)==i)-1)+1:find(m(1,:)==i)*n);
%利用 h 表示最後的 huffman 編碼
    l1(i)=length(find(abs(h(i,:))~=32));
%計算每一個編碼的長度(用於求平均碼長)
end
l=sum(p.*l1);%計算平均碼長
```

3. 範例(P138,problem4.2)

Illustrative Problem 4.2 [Huffman Coding] A discrete-memoryless information source with alphabet

$$\mathcal{X} = \{x_1, x_2, \dots, x_6\}$$

and the corresponding probabilities

$$\mathbf{p} = \{0.1, 0.3, 0.05, 0.09, 0.21, 0.25\}$$

is to be encoded using Huffman coding.

1. Determine the entropy of the source.
2. Find a Huffman code for the source and determine the efficiency of the Huffman code.

第一題用到的 function 為 entropy

Entropy 簡單來說是衰退的指標

假設有很多數值要取值並且每種取值得占比都不一樣則此函數為

$$-\sum(p(i).*\log_2[p(i)])$$

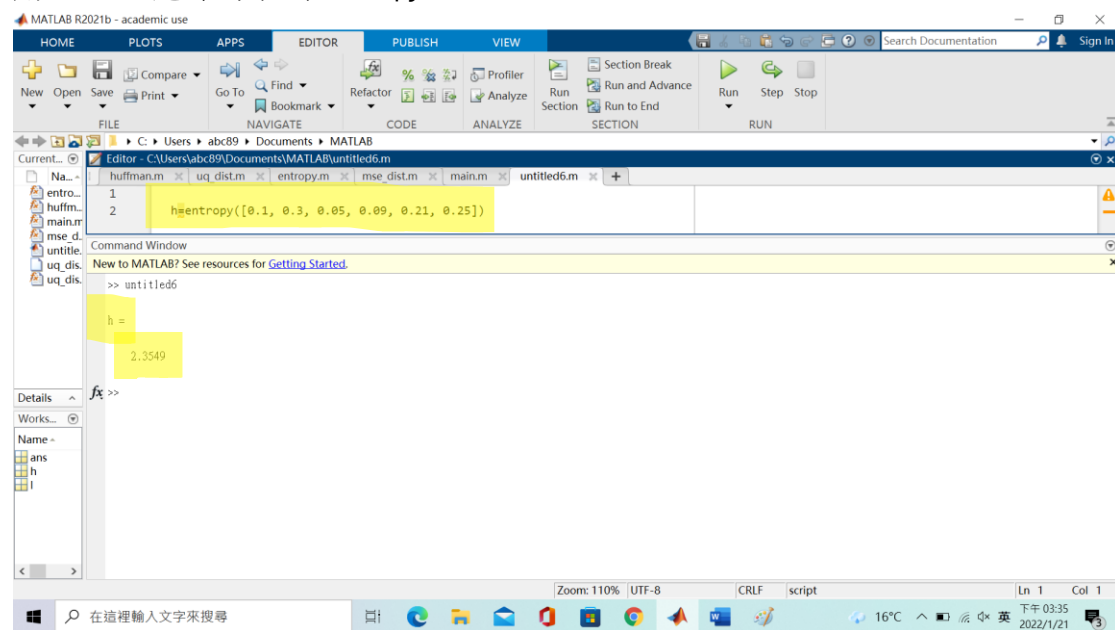
就是對每個 p 求其對應的 $p(i).*\log_2[p(i)]$ ，然後再把這些所有結果加起來再取相反數。

Matlab function(entropy)

```
function h=entropy(p)
% H=ENTROPY(P) returns the entropy function of
% the probability vector P.
if length(find(p<0))~=0,%機率不可小於 0
    error('Not a prob. vector, negative component(s)')
end
if abs(sum(p)-1)>10e-10,%機率和不可大於 1
    error('Not a prob. vector, components do not add up to 1')
end
h=sum(-p.*log2(p));%entropy 公式
```

回到題目的問題一=> Determine the entropy of the source.

用 matlab 跑即可求出 entropy=2.3549



第二題要求求出 Huffman code 的編碼效率

思路=>首先求出 huffman code 的平均碼長,再去用第一題求出的 entropy 值除以平均碼長即可求出編碼效率

結果(平均碼長為 2.38)

The screenshot shows the MATLAB R2021b interface. In the Command Window, the following code is executed:

```
h = entropy([0.1, 0.3, 0.05, 0.09, 0.21, 0.25])
[h, l] = huffman([0.1, 0.3, 0.05, 0.09, 0.21, 0.25])
```

The output for `h` is a 6x6 char array:

```
h =
    ' 010'
    ' 11'
    ' 0110'
    ' 0111'
    ' 00'
    ' 10'
```

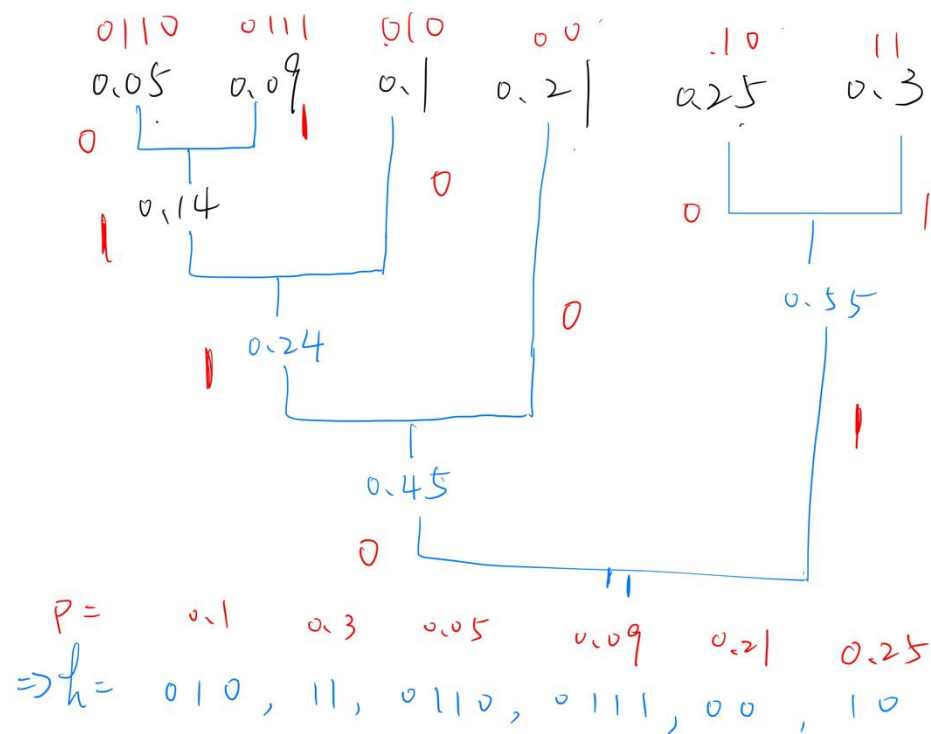
The output for `l` is:

```
l =
    2.3800
```

即可馬上算出

$$\eta = \frac{2.3549}{2.38} = 0.9895 \Rightarrow 98.95\%$$

以下為計算過程



`h` 就是 matlab 執行的每個值對應的編碼

=>0110,0111,010,00,10,11(跟課本上的答案一模一樣)