

Let the source, X , have letters $\{1,2,3,4\}$ and probabilities 0.5, 0.25, 0.125, 0.125. Design a Huffman code for this source. Compare the average codeword length to the source entropy. ($H(X) = 1.75$ bits.)

```
>> p=[.5 .25 .125 .125];
>> [dict,length]=huffmandict([1:4],p) % Design the Huffman code.
dict =
    [1] [      0]
    [2] [1x2 double]
    [3] [1x3 double]
    [4] [1x3 double]

length =
    1.7500          % The average codeword length equals the entropy.

>> dict{:,2} % Use braces, { } to list the binary codewords.
ans =
    0
ans =
    1    0
ans =
    1    1    1
ans =
    1    1    0

% Generate a sequence of source letters; encode the source as a bitstream, decode the
% bitstream.

>> signal = [1 4 2 1 1 3 2 2 1]; % Generate a sequence of source symbols (letters).

>> bitstream=huffmanenco(signal,dict) % Encode the source using the Huffman code.
bitstream =
Columns 1 through 14
    0    1    1    0    1    0    0    0    1    1    1    1    0    1
Columns 15 through 16
    0    0

>> decoded_signal=huffmandeco(bitstream,dict) % Decode the bitstream.
decoded_signal =
    1    4    2    1    1    3    2    2    1
```