

```

% Created on 13/02/25
% Created by Sahastrajit B., BT22ECE093
% Fourth Practical ~ Implementation of Shannon-Fano Coding.

clc
clear all close
all

% Creating Symbols, their Probabilities and Input String
symbols = ['P', 'Q', 'R', 'S', 'T'];
probabilities = [0.08, 0.36, 0.22, 0.22, 0.12];
str_example = 'PTRQS';

% Arranging in descending order
[sorted_prob, index] = sort(probabilities, 'descend');
sorted_symbols = symbols(index);

% Creating empty Codewords for all Symbols
codewords = cell(size(sorted_prob));
codewords(:) = {' '};

% Assigning codewords
codewords = assign_codes(codewords,sorted_prob);

% Rearranging Codewords according to given list of Symbols
[~, reverse_idx] = sort(index);
codes = codewords(reverse_idx);

% Encoding and Decoding of given Input String
encoded_string = sf_encode(str_example,symbols,codes);
decoded_string = sf_decode(encoded_string,symbols,codes);

% Printing Output~
fprintf('\n Symbol | Probability | Shannon-Fano Code |\n');

for i = 1:length(symbols)
    fprintf('    %c      |    %.2f      |    %s      |\n', ...
        symbols(i), probabilities(i), codes{i});
end

fprintf('\n\nOriginal String: %s\n', str_example);
fprintf('=> Encoded String: %s\n', encoded_string);
fprintf('=> Decoded String: %s\n', decoded_string);

function [codes] = assign_codes(codes,probabilities)

    if(length(probabilities)==1)
        return;
    end

    % Even Partition into two groups
    number = length(probabilities);
    total = sum(probabilities);
    temp = 0;

```

```

for ii = 1:number
    temp = temp + probabilities(ii);
    diff(ii) = temp-(total-temp);
end
[~,split_i] = min(abs(diff));

% Assignment of single bit 0 or 1
for jj = 1:split_i
    codes{jj} = [codes{jj} '0'];
end
for jj = (split_i+1):number
    codes{jj} = [codes{jj} '1'];
end

% Further Assignment (Recursive function call)
codes(1:split_i) = assign_codes(codes(1:split_i), ...
    probabilities(1:split_i));
codes((split_i+1):number) = assign_codes(codes((split_i+1):number), ...
    probabilities((split_i+1):number));

end

function [en_str] = sf_encode(in_str,symbols,codes)
    en_str = '';
    for ii = 1:length(in_str)
        for jj = 1:length(symbols)
            % Updating Encoded String according to Matched Symbols
            if(in_str(ii)==symbols(jj))
                en_str = [en_str,codes{jj}];
            end
        end
    end
end

function [de_str] = sf_decode(en_str,symbols,codes)
    de_str = '';
    temp = '';
    for ii = 1:length(en_str)
        temp = [temp en_str(ii)];
        for jj = 1:length(codes)
            % If codeword matches, Decoding symbols for string
            if(strcmp(temp,codes{jj}))
                de_str = [de_str,symbols(jj)];
                temp = '';
            end
        end
    end
end

```

Symbol	Probability	Shannon-Fano Code
P	0.08	111
Q	0.36	00
R	0.22	01
S	0.22	10
T	0.12	110

Original String: PTRQS
=> Encoded String: 111110010010
=> Decoded String: PTRQS

Published with MATLAB® R2024b