Huffman coding code

% Huffman coding

% finding codes and compression ratio

close all;

clear all;

P\_x = [0.05 0.2 0.15 0.4 0.1 0.1];

x = {1; 2; 3; 4; 5; 6};

no\_sym = length(x);

y = x;

p\_y = P\_x;

% Create a cell array for codes

codes = cell(1, no\_sym);

% Arrange in descending order

for i = 2:no\_sym

for j = i:-1:2

if p\_y(j) < p\_y(j - 1)

break;

else

temp = p\_y(j - 1);

p\_y(j - 1) = p\_y(j);

p\_y(j) = temp;

temp2 = y(j - 1);

y(j - 1) = y(j);

y(j) = temp2;

end

end

end

la = length(y);

i = 1;

while la >= 2

for j = 1:length(y{la})

index = y{la}(j);

codes{index} = [1, codes{index}];

end

for j = 1:length(y{la - 1})

index = y{la - 1}(j);

codes{index} = [0, codes{index}];

end

p\_y(la - 1) = p\_y(la) + p\_y(la - 1);

p\_y = p\_y(1:la - 1);

y{la - 1} = [y{la - 1}, y{la}];

y = y(1:la - 1);

for j = la - 1:-1:2

if p\_y(j) < p\_y(j - 1)

break;

else

temp = p\_y(j);

p\_y(j) = p\_y(j - 1);

p\_y(j - 1) = temp;

temp2 = y(j);

y(j) = y(j - 1);

y(j - 1) = temp2;

end

end

la = length(y);

end

% Display Huffman codes

for i = 1:no\_sym

fprintf('Symbol %d Huffman code: ', i);

disp(codes{i});

end

% Calculate and display the compression ratio

bit\_count\_original = no\_sym \* 8; % Assuming 8 bits per symbol

bit\_count\_compressed = 0;

for i = 1:no\_sym

bit\_count\_compressed = bit\_count\_compressed + length(codes{i});

end

compression\_ratio = bit\_count\_original / bit\_count\_compressed;

fprintf('Compression ratio: %.2f\n', compression\_ratio);