In [ ]:
```
import numpy as np
from itertools import product
```

In [ ]:
```
# Problem 1.33

def prob33(p):
    l = np.ceil(np.log2(1/p))
    return sum(p*l), l
```

In [ ]:
```
def prob34(p, n):
    p_n = np.product(sorted(list(product([p, 1-p], repeat=n))), axis=1)
    return prob33(p_n)[0]
```

In [ ]:
```
p = .8
H = p * np.log2(1/p) + (1-p) * np.log2(1/(1-p))
for n in range(1,16):
    print(prob34(p,n)/n, H <= prob34(p,n)/n and prob34(p,n)/n <= H + 1)
```

```
1.4 True
0.8999999999999999 True
0.7333333333333331 True
0.9 True
0.7999999999999997 True
0.7333333333333334 True
0.8285714285714281 True
0.7749999999999992 True
0.7333333333333347 True
0.8000000000000009 True
0.763636363636365 True
0.7333333333333392 True
0.7846153846153777 True
0.7571428571428147 True
0.7333333333333686 True
```

1.31. Show that the average word length of the encoding given in (1.33) is 0.728.

$$\frac{(0.512)(1)+\left[(0.8)^2(.2)(3)\right]3+3\left[(0.8)(0.2)^2(5)\right]+(0.008)(5)}{3}$$

$$\frac{awl(C,P)}{N} = \boxed{0.728}$$

where N is the number of symbols per word

1.32. Perform the encoding method given in the proof of Theorem 1.6.4 for the cases in Examples 1.6.5 and 1.6.6. In addition to determining the encodings, draw the binary trees corresponding to the encodings.

Example 1.6.5

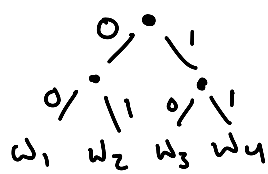
$$w_1 = 0$$
$$w_2 = 2^{2-2} = 2^0 = 1$$
$$w_3 = 2^{2-2} + 2^{2-2} = 2^0 + 2^0 = 2$$
$$w_4 = 2^{2-2} + 2^{2-2} + 2^{2-2} = 2^0 + 2^0 + 2^0 = 3$$

$w_1 \to 00$

$w_2 \to 01$

$w_3 \to 10$

$w_4 \to 11$



Example 1.6.6

$$W_1 = 0$$

$$W_2 = 2^{2-2} = 2^0 = 1$$

$$W_3 = 2^{2-2} + 2^{2-2} = 2^0 + 2^0 = 2$$

$$W_4 = 2^{3-2} + 2^{3-2} + 2^{3-2} = 2+2+2 = 6$$

$$W_5 = 2^{3-3} + 2^{3-2} + 2^{3-2} + 2^{3-2} = 7$$

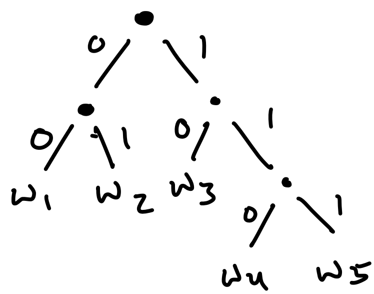
$W_1 \rightarrow 00$

$W_2 \rightarrow 01$

$W_3 \rightarrow 10$

$W_4 \rightarrow 110$

$W_5 \rightarrow 111$



1.35. Let $H(p_1, \ldots, p_n)$ denote the entropy of a random variable with support $\{x_1, \ldots, x_n\}$ and probabilities $p_k = P(X = x_k)$. The entropy depends only on the probabilities, so this makes sense.

(i) Show that for any $\lambda \in (0,1)$ and any choice of $p_1, \ldots, p_n > 0$ with $\sum p_k = 1$ the following relation holds:

$$H(p_1, \ldots, p_{n-1}, \lambda p_n, (1-\lambda)p_n) = H(p_1, \ldots, p_n) + p_n H(\lambda, 1-\lambda). \quad (1.70)$$

(ii) Justify the identity (1.70) by discussing how a signal from a source alphabet S of $n+1$ symbols can be compressed by first combining the last two symbols into one symbol and compressing, and then following up with a second compressed binary signal to differentiate which of the last two symbols was represented at each point in the original signal.

i) $\forall \lambda \in (0,1)$ and $p_1, \ldots, p_n > 0$ $\sum p_k = 1$

By 1.39 we know

$$H(p_1, \ldots, p_{n-1}, \lambda p_n, (1-\lambda) p_n)$$

$$= \sum_{k=1}^{n-1} p_k \log_2 \frac{1}{p_k} + \lambda p_n \log_2 \frac{1}{\lambda p_n} + (1-\lambda) p_n \log_2 \frac{1}{(1-\lambda) p_n}$$

$$= \sum_{k=1}^{n-1} p_k \log_2 \frac{1}{p_k} + \lambda p_n \log_2 \frac{1}{\lambda} + \lambda p_n \log_2 \frac{1}{p_n} + p_n \log_2 \frac{1}{(1-\lambda) p_n} - \lambda p_n \log_2 \frac{1}{(1-\lambda) p_n}$$

Note $(1-\lambda) p_n \log_2 \frac{1}{(1-\lambda) p_n} = p_n \log_2 \frac{1}{(1-\lambda) p_n} - \lambda p_n \log_2 \frac{1}{(1-\lambda) p_n}$

$$= p_n \log_2 \frac{1}{p_n} + p_n \log_2 \frac{1}{1-\lambda} - \lambda p_n \log_2 \frac{1}{1-\lambda} - \lambda p_n \log_2 \frac{1}{p_n}$$

Thus our original equation

$$\sum_{k=1}^{n-1} p_k \log_2 \frac{1}{p_k} + \lambda p_n \log_2 \frac{1}{\lambda} + \cancel{\lambda p_n \log_2 \frac{1}{p_n}} + p_n \log_2 \frac{1}{p_n} + p_n \log_2 \frac{1}{1-\lambda} - \lambda p_n \log_2 \frac{1}{1-\lambda} - \cancel{\lambda p_n \log_2 \frac{1}{p_n}}$$

$$= \sum_{k=1}^{n} p_k \log_2 \frac{1}{p_k} + \lambda p_n \log_2 \frac{1}{\lambda} + (1-\lambda) p_n \log_2 \frac{1}{1-\lambda}$$

$$= H(p_1, \ldots, p_n) + p_n H(\lambda, (1-\lambda))$$

ii) If the last two symbols from the alphabet are combined then you would need an additional compression to help differentiate the last symbol of the compressed alphabet and thus you would have additional entropy to differentiate.