# 2nd Homework: Networking

Due Saturday, October 2, 1 second before midnight
typeset and PDF only

This problem set is about the Hamming code presented in lecture. It is known as Hamming 15,11, but the details are different.

30 pts **Problem 1:**

The bitwise binary sum of two encodings is again an encoding. We check that something is an encoding by multiplying with the parity check matrix $H$ and verifying that the result is the zero vector. Calculate the encoding of 10 pairs of vectors and verify that the bitwise binary sum is again an encoding.

Example:
$m_1 = [0,0,1,0,0,1,0,0,1,0,1]$
$m_2 = [1,1,0,0,0,1,1,0,0,1,1]$
$\text{encoded}_1 = [0,0,1,0,0,1,0,0,1,0,1,1,1,1,0]$
$\text{encoded}_2 = [1,1,0,0,0,1,1,0,0,1,1,0,1,1,0]$
$\text{encoded}_1 \oplus \text{encoded}_2 = [1,1,1,0,0,0,1,0,1,1,0,1,0,0,0]$
$H \cdot (\text{encoded}_1 \oplus \text{encoded}_2) = (0,0,0,0)$
A little bit of algebra shows that $\text{encoded}_1 \oplus \text{encoded}_2 = G \cdot (m_1 \oplus m_2)$

If you know Python programming: You can just provide the Python code that checks this property. In this case, the following simple integer to binary array transformation function might be useful:

```
def int2bits(num, length):
    result = np.zeros(length).astype(int, copy = False)
    for i in range(length):
        if num%2:
            result[i]=1
        num = num//2
    return result
```

30 pts **Problem 2:**

The minimum Hamming weight of the code words in this code is at least 3. This exclude the code word consisting only of zeroes. Generate 10 random messages and verify this for these 10 messages.

If you know Python programming, you might prefer to just submit a Python program that checks that the only code word that has weight less than 3 is the zero codeword.

40 pts **Problem 3:**

The Hamming code is guaranteed to discover one error and to detect two errors.

Assume that the message [1 0 0 0 0 0 0 1 1 1 0 1 1 0 0] is being sent but [1 0 1 1 0 0 0 1 1 1 0 1 1 0 1] is received. What happens?
Assume that the message [1 0 0 0 0 0 0 1 1 1 0 1 1 0 0] is being sent but [1 0 0 0 1 1 0 1 1 1 0 1 1 1 0] is received. What happens?