Applications of polynomials

(exercises)

Application of discrete models

1. Implement the Lagrange interpolation, i.e. write a function that gets $n+1$ points in a list: $[(x_1, y_1), (x_2, y_2), \ldots, (x_{n+1}, y_{n+1})]$ (where x_i's are all different), and returns the polynomial of degree $\leq n$ for which $p(x_1) = y_1$, $p(x_2) = y_2$, $\ldots$, $p(x_{n+1}) = y_{n+1}$. Don't use the interpolation functions of Sage (other than for checking).

2. Simulate Shamir's secret sharing as follows:
   a) Write a function that receives the parameters n, k, S, and it generates the n secret shares. Return value: a list of length n containing the (x_i, y_i) points.
   b) Write a function that receives k of the above n shares (a list of length k of (x_i, y_i) points), and reconstructs the secret S.

3. Implement CRC as follows, using the generator polynomial $g = x^4 + x + 1 \in \mathbb{Z}_2[x]$.
   a) Write a function that gets a message m (a bit sequence), and calculates the transmitted codeword, i.e. it extends m with the CRC check value. You can use Sage's polynomial operations, but the input and output are bit sequences, i.e. list of 0's and 1's, not polynomials.
   b) Write a function that checks the received codeword, and returns whether it is valid.
   c) Generate a random 16-bit message, and try function a) on it, then check the resulting codeword using b).
   d) Change (corrupt) the codeword and check that too.
   e) Change the codeword in such a way that it will be accepted by the check.

4. a) Write a function that performs the *SubBytes* transformation of AES for a given byte. The input and output are integers (between 0 and 255).
   b) Print all 256 results in a 16×16 table (in row major order). Check the result using the Internet (search for "Rijndael S-box").
   c) Is this transformation invertible? (I.e. do all possible values occur exactly once?)
   d) Does this transformation have a fixed point? (I.e. a value whose image is itself?)