1 References for the Course

1. Modern Computer Algebra by Von Zur Gathen and Jurgen Gerhard.
2. Introduction to Computational Number Theory by Victor Shoup (Available on author’s website)
3. Writeups and papers on the web.

2 Basic Operations

2.1 1. In Number Theory

a. Addition, subtraction.
b. Multiplication, division, modulo, factorization, testing primality.
c. LCM, GCD.

2.2 2. In Algebra

2.2.1 A. Linear Algebra

b. Solving system of linear equations.
c. Finding eigen values.

2.2.2 B. Polynomial Algebra

a. Addition, Multiplication.
b. Finding roots, factorization.
c. Checking irreducibility.

2.2.3 C. Abstract Algebra

a. Finding inverse of a group element.
b. Finding non-trivial homomorphisms.
c. Finding a generator set in a group.
d. Finding cosets.
e. Finding order of an element.

3 Tools for Designing Algorithms

2. Fast Fourier Transform.
3. Divide and Conquer Technique.
4. Short vectors in a Lattice.
5. Elliptical curves.

4 Reed-Solomon Code (Error correcting code)

Let $I$ be the entire data to be stored in a cd. Break $I$ into chunks of $b \times k$ bits.
Let $F$ be the finite field of size $2^b$.
Each chunk is coded separately.

4.1 Coding one chunk of $b \times k$ bits

Break the chunk into $k$ blocks of $b$ bits.
Let these be $d_0, d_1, \ldots, d_{k-1}$
Treat each $d_i$ as an element of $F$
Let polynomial $P(x) = \sum_{i=0}^{k-1} d_i \times x^i$
Let $e_0, e_1, \ldots, e_{n-1}$ be $n$ distinct elements of $F, n \leq 2^b$
Let $f_j = P(e_j)$
Then codeword corresponding to $d_0, d_1, \ldots, d_{k-1}$ ($k \times b$ bits) is $f_0, f_1, \ldots, f_{n-1}$ ($n \times b$ bits)
Note that $n > k$. This is where the redundancy comes in.