TD1 : Algorithms and practical complexity

1 Key points

- The practical complexity of elementary operations in $\mathbb{Z}$ and $K[x]$,
- The definition of gcd and lcm. Bezout's theorem,
- Be familiar with extended Euclidean algorithm for both integers and polynomials,
- Be able to analyse an elementary algorithm e.g. a sorting algorithm or a simple algorithm in graph theory,
- Know the elementary algorithms for the ring $(\mathbb{Z}/N\mathbb{Z}, +, \times)$.

2 Experimental study of some running times

Using the computer's clock, study the practical complexity (the running time as a function of the input size) for the following operations

1. addition of two integers,
2. multiplication of two integers,
3. computing $a^b \bmod c$ where a, b and c are integers,
4. multiplication of two square matrices with coefficients in $\mathbb{Z}/n\mathbb{Z}$.

If one guesses a complexity function like

$$T(n) = u + vn^{\alpha},$$

one may try to pin down the constants u, v and w as accurately as possible.

One may also try to evaluate the complexity of some function of one's prefered computer algebra system, e.g. the PARI/GP functions **isprime**, **factor**, **nextprime**, **vecsort**.

3 Sorting

Implement a slow sorting algorithm, implement a fast sorting algorithm, compare their practical complexities.

4 The group $(\mathbb{Z}/N\mathbb{Z})^*$

Implement an algorithm that on input a prime integer N returns a generator of $(\mathbb{Z}/N\mathbb{Z})^*$. What is the complexity of this algorithm ? What is the hardest step in this algorithm ?

5 Graphs

One is given a non-oriented graph (E, V) where $V \subset E \times E$.

Implement an algorithm to compute the connected component of a vertex.