

MA810. Computer Algebra. Project 2

Factoring integers

November 7, 2008

1. Implement the Sieve of Eratosthenes to determine all primes $p < 2^{20}$.
2. Implement the Trial division algorithm (Algorithm 19.1, page 533) for prime factors $p < 2^{20}$. Use the generated list as an array of primes.
3. Implement Pollard's ρ method (Algorithm, 19.8, page 538). Please allow exchange or modification of the iteration function. Make sure that your algorithm is correct. How does the performance of Pollard's ρ algorithm in terms of running time and finding prime factors changes if we use $x_{i+1} = x_i^2 + x_i + 1 \pmod N$ or $x_{i+1} = x_i^3 + 1 \pmod N$ as iteration function instead of the suggested iteration function $x_{i+1} = f(x_i) = x_i^2 + 1 \pmod N$. Use the program "cs810a_rand" to generate 100 random numbers with bit length of 40, 50, and 60 bit.
4. Implement Lenstra's elliptic curve method (Algorithm 19.22, page 553). Make sure that your algorithm is correct.
5. Develop a factorization strategy for integers up to 128 bit. In order to develop a strategy we use composite numbers $N = p \cdot q$, with p and q prime to determine the performance of each algorithm in finding primes of a certain size. We use $|p|$ and $|q|$ to denote the bit length of p and q .
 - (a) Use the program "cs810a_fact" to generate integers $N = p \cdot q$ of bit length 200. Generate 100 numbers for each $|p| \in \{10, 20, 30, 40, 50, 60\}$ and $|q| = 200 - |p|$.
 - (b) Test the performance of your factorization algorithms (Trial division, Pollard's ρ and Lenstra's elliptic curve) with the above created numbers.
 - (c) Decide on a strategy based upon your findings.

Please use the method described in Project 1 to measure the performance of the algorithms.

This project is due by November 21.