Factoring large integers using Quadratic Sieve

Christopher Teljstedt
chte@kth.se

Henrik Viksten
hviksten@kth.se

November 10, 2013

Abstract

Suspendisse id urna vel risus venenatis ultrices ut vel odio. Donec aliquet est at magna tincidunt ut rutrum lacus cursus. Praesent ultricies aliquam erat quis scelerisque. Vestibulum interdum interdum augue, at placerat turpis tempus nec. Vestibulum feugiat, tellus ultrices tempor fermentum, ipsum dolor vestibulum eros, sed vulputate felis eros eget ipsum. Fusce ultricies dapibus turpis non pretium. Suspendisse potenti. Integer porttitor, lorem ac mattis fermentum, metus neque scelerisque sapien, vel lobortis orci erat at sapien.

I. Introduction

This report will cover the work of a integer factorization program written in the course DD2440 advanced algorithms. Methods that can be used to solve the integer factorization problem in a effective way using existing algorithms and an explanation of methods used.

I.1 Purpose

The goal of the program is to pass kattis test cases with as high score as possible, this is done by improving the program step-by-step and gradually increasing the performance. In the report the methods used will be analyzed. How they work independently and their correlation in our implementation.

I.2 Problem

An unknown set of 100 integers in varying size are used as input to the algorithm from kattis. The output is either the factors of the input in case it is solved or the string fail otherwise. One of the problems is dealing with large integers within the timeframe but also finding every single non-trivial factor. The restrictions in kattis are 15 seconds and 64MB of memory.

I.3 Scope

There is no intention of reinventing any methods that already exists, the idea is to create a solver that can factorize unknown integers and in the end get a high score on kattis.

I.4 Statement of Collaboration

Some text.

II. Preliminaries

Notation. By '$\log_b$' we denote the base b logarithm and the natural logarithm denotes by $\ln = \log_e$ with $e \approx 2.71828$. The largest integer $\leq x$ is denoted by '[x]'. The number of primes $\leq x$ is denoted by '$\pi(x)$', and due the *Prime number theorem*[1] we know that $\pi(x) \approx x/\ln(x)$.

Modular arithmetic. Throughout this paper '$x \equiv y \bmod n$' means that $(x-y)$ is a multiple of n whereas x, y and $n \in \mathbb{N}_{\neq 0}$. Similarly, '$x \not\equiv y \bmod n$' mean that $(x-y)$ is not a multiple of n. Euclid's algorithm for finding the *greatest common divisor* of two non-negative integers, say x and z, is denoted by $\gcd(x, z)$.

III. Quadratic Sieve

III.1 Brief explaination

The quadratic sieve algorithm is today the algorithm of choice when factoring very large composite numbers with no small factors. The general idea behind quadratic factoring is based on Fermat's observation that a composite number if one can find two find two integers $x, y \in Z$, such that $x^2 \equiv y^2 \pmod{n}$ and $x \not\equiv \pm y \pmod{n}$. This would imply that,

$$n \mid x^2 - y^2 = (x-y) \cdot (x+y) \tag{1}$$

but n neither divides $(x-y)$ nor $(x+y)$. Furthermore it can be rewritten as $(x-y) \cdot (x+y) = k \cdot p \cdot q$ for some integer k, thus becoming two possible cases.

- either p divides $(x-y)$ and q divides $(x+y)$, or vice versa.
- or both p and q divides $(x-y)$ and neither of them divides $(x+y)$, or vice versa.

Hence, the greatest common divisor of $(x-y, n)$ and $(x+y, n)$ would with with a $1/2$ probability yield first case which is p or q and a non-trivial factor of n is found. In the second case we get n or 1 and trivial solution is found.

Carl Pomerance suggested a method to find these tsuch squares[2]. The first step in doing so is to is to define the polynomial

$$Q(x) = (x + \lfloor \sqrt{s} \rfloor)^2 - N \tag{2}$$

References

[1] G.H. Hardy, E.M. Wright, D.R. Heath-Brown, and J. Silverman. *An Introduction to the Theory of Numbers*. Oxford mathematics. OUP Oxford, 2008.

[2] Carl Pomerance. The quadratic sieve factoring algorithm. In Thomas Beth, Norbert Cot, and Ingemar Ingemarsson, editors, *Advances in Cryptology*, volume 209 of *Lecture Notes in Computer Science*, pages 169–182. Springer Berlin Heidelberg, 1985.