Lecture 20

Lecturers: Markus Bläser, Chandan Saha *Scribe: Chandan Saha*

Today, we will discuss another application of smooth numbers, namely:

- The Quadratic Sieve method for factoring integers.

1 The Quadratic Sieve method

The Quadratic Sieve (QS), which was first proposed by Pomerance [Pom82, Pom84], is also based on Kraitchik's scheme of finding 'distinct' roots of a square modulo N. But unlike Dixon's algorithm, this method is deterministic with a heuristic analysis showing a time bound of $e^{(1+o(1))\sqrt{\ln N \ln\ln N}}$ operations. Although it has the same asymptotic complexity as the best (rigorous) randomized algorithm, the QS is much more efficient in practice. As for the assumptions made in the analysis of the QS, in Pomerance's [Pom08] own words "...perhaps we should be more concerned with what is *true* rather than what is *provable*, at least for the design of a practical algorithm.".

The Quadratic Sieve method generates a sequence of squares modulo N using the polynomial $x^2 - N$, by varying integer x from $\sqrt{N}$ to $\sqrt{N} + N^{o(1)}$. We could say $x^2 \bmod N$ instead of $x^2 - N$, they being the same as $x \leq \sqrt{N} + N^{o(1)}$. This step of deterministically generating squares modulo N is in contrast to Dixon's algorithm where x is chosen randomly and then $x^2 \bmod N$ is computed. As before, we are interested in those numbers in the sequence that are y-smooth (for some fixed y). Understanding the distribution of smooth numbers in this sequence is a difficult number theoretic problem. Instead, to make the analysis go through we assume that the sequence generates y-smooth numbers in the same frequency as numbers picked randomly from the range $[0, 2N^{\frac{1}{2}+o(1)}]$. Since $\sqrt{N} \leq x \leq \sqrt{N} + N^{o(1)}$, $x^2 - N$ is between 0 and roughly $X = 2N^{\frac{1}{2}+o(1)}$. In other words, we assume that a y-smooth number is encountered after about $\frac{X}{\psi(X,y)}$ numbers of the sequence $\{x^2 - N\}$ as x is varied from $\sqrt{N}$ to $\sqrt{N} + N^{o(1)}$. This is where the Quadratic Sieve becomes heuristic in nature. Although there is no rigorous theory to back this assumption, years of experimental analysis have strongly supported it. So let us proceed with it and see the step to which the QS owes both its name and its efficiency.

Recall that, in Dixon's algorithm, we test for y-smoothness of $x^2 \bmod N$ by trial division with all the k primes less than y. This takes $O((k+n)\mathsf{M}_\mathsf{I}(n))$ time per number, where $n = \ln N$. We cannot do better than this if we test for smoothness, one number at a time. But in our case the numbers originate from a well-defined sequence. One can hope to spend much less overall time by testing smoothness of a collection of numbers together. To draw the analogy, let us briefly visit the Sieve of Eratosthenes method.

Suppose we want to find all the primes less than B. If we sequentially test for primality of all the B numbers, we end up spending $\tilde{O}(B \log^2 B)$ time (using Miller-Rabin primality test). Instead, the Sieve of Eratosthenes starts with an array of size B with all entries initialized as 'unmarked'. At each step, the process finds the next 'unmarked' index $p > 1$ and 'marks' those entries that are higher multiples of p. In the end, only the prime indices remain 'unmarked'. The total time spent is $\sum_{\text{prime } p < B} \frac{B}{p} \log B$, where the $\log B$ factor is due to the increment of the counter by p for each subsequent 'marking'. Using the analytic bound $\sum_{\text{prime } p < B} \frac{1}{p} = O(\log\log B)$, the time complexity comes out to be $O(B \log B \log\log B)$. By finding the primes together, the Sieve spends $O(\log B \log\log B)$ operations per number on average, as opposed to $\tilde{O}(\log^2 B)$ operations for primality testing. Pomerance made the observation that this idea of sieving primes can be adapted to sieve smooth numbers from the sequence $\{x^2 - N\}$.

Let p be a prime less than y and $p \nmid N$. Then p divides $x^2 - N$ if x is $\pm c$ modulo p where c is a square root of N modulo p. Since $x^2 - N$ is also an element of the sequence $\{x^2 - N\}$, $x = \lfloor \sqrt{N} \rfloor + i$ for some index $0 \leq i \leq N^{o(1)}$, hence there are two fixed values $a, b < p$ such that $x^2 - N$ is divisible by p if and only if $i = a$ or $b \bmod p$. This means, if the sequence $\{x^2 - N\}$ is presented as an array with the i^{th} index containing $(\lfloor \sqrt{N} \rfloor + i)^2 - N$, then the numbers that are divisible by p are exactly placed at the indices $\{a + kp\}_{k \geq 0}$ and $\{b + kp\}_{k \geq 0}$.

A sieving process can start from index a (similarly, b) and increment the counter by p, iteratively, to get to the numbers in the sequence that are divisible by p. A number that is divisible by p is then replaced by the quotient obtained by dividing it by the highest power of p. After this sieving is done for every prime $p < y$, all those numbers that have changed to 1 are y-smooth. If B is the size of the array then the total sieving time is, $O(\sum_{\text{prime } p < y} \frac{B}{p} \cdot n \cdot \mathsf{M}_{\mathsf{I}}(n)) = O(B \cdot \log\log y \cdot n \cdot \mathsf{M}_{\mathsf{I}}(n))$, where $n = \log N$. How large should B be? Since there are k primes below y, just like Dixon's algorithm, we need $k + 1$ y-smooth numbers. We have assumed before that one y-smooth number is present in (roughly) every $\frac{X}{\psi(X,y)}$ numbers of the sequence, where $X = 2N^{\frac{1}{2}+o(1)}$. Therefore, it is sufficient if we take $B \approx (k+1) \cdot \frac{X}{\psi(X,y)}$ for the analysis. The value of y is fixed in the analysis of the following algorithm.

Algorithm 1 Quadratic Sieve method

1. Find all primes, $p_1, \ldots, p_k \leq y$. If $p_i | N$ for any $1 \leq i \leq k$, return p_i.
2. Sieve the sequence $(\lfloor \sqrt{N} \rfloor + j)^2 - N$, $0 \leq j \leq B$, for $(k+1)$ y-smooth numbers, $\gamma_i = \alpha_i^2 - N = p_1^{e_{i1}} \ldots p_k^{e_{ik}}$, $1 \leq i \leq k+1$. Let $v_{\gamma_i} = (e_{i1} \mod 2, \ldots, e_{ik} \mod 2)$.
3. Find $I \subset \{1, \ldots, k+1\}$ such that $\sum_{i \in I} v_{\gamma_i} = 0$ over $\mathbb{F}_2$.
4. Let $\gamma = \prod_{i \in I} \gamma_i = p_1^{e_1} \ldots p_k^{e_k}$, $\beta = p_1^{e_1/2} \ldots p_k^{e_k/2} \mod N$ and $\alpha = \prod_{i \in I} \alpha_i \mod N$.
5. If $\gcd(\alpha + \beta, N) \neq 1$ return the factor.

Heuristic time complexity - As with Dixon's algorithm, the time complexity of the Quadratic Sieve is mainly contributed by step 2 and step 3. If $y = X^{\frac{1}{u}}$ then by Lemma 4 in lecture 18, $\frac{X}{\psi(X,y)} \approx u^u$. So the total time spent in step 2 is heuristically, $O(B \cdot \log\log y \cdot n \cdot \mathsf{M}_{\mathsf{I}}(n)) = \tilde{O}(B) = \tilde{O}(k \cdot u^u) = \tilde{O}(X^{\frac{1}{u}} u^u)$, ignoring lower order terms in N. This expression is minimized when $u \approx \sqrt{\frac{2 \ln X}{\ln\ln X}}$, implying that $y = e^{(2^{-1/2}+o(1))\sqrt{\ln X \ln\ln X}}$. Taking $X = 2N^{\frac{1}{2}+o(1)}$, the time complexity of step 2 comes out to be $O(X^{\frac{1}{u}} u^u) = e^{(1+o(1))\sqrt{\ln N \ln\ln N}}$. Notice that the vector v_{γ_i} (in step 2) is fairly sparse, it has at most $\log N$ nonzero entries. So the matrix formed by the $k+1$ vectors, $v_{\gamma_1}, \ldots, v_{\gamma_{k+1}}$, contains at most $(k+1)\log N$ nonzero entries. Using Wiedemann's sparse matrix method [Wie86], step 3 can be implemented in $O((k+1) \cdot (k+1)\log N)) = e^{(1+o(1))\sqrt{\ln N \ln\ln N}}$ time.

Note that in step 5, $\gcd(\alpha + \beta, N)$ can possibly be 1, but since α and β are computed 'differently', it is 'likely' that $\alpha \neq \pm\beta \mod N$, in which case we indeed get a non-trivial factor. This is yet another reason why the analysis of the algorithm is heuristic in nature.

For an excellent survey on the Quadratic Sieve and how it evolved from earlier factoring methods, refer to the articles [Pom08], [Pom96] and [Pom84] by Carl Pomerance. Several enhancements of the QS are described in chapter 6 of [CP05].

References

[CP05] Richard Crandall and Carl Pomerance. *Prime Numbers: A Computational Perspective*. Springer, USA, 2005.

[Pom82] Carl Pomerance. Analysis and comparison of some integer factoring algorithms. In H.W. Lenstra Jr. and R. Tijdeman, editors, *Computational Methods in Number Theory*, volume 154 of *Math. Centrum Tract*, pages 89–139, 1982.

[Pom84] Carl Pomerance. The Quadratic Sieve Factoring Algorithm. In *EUROCRYPT*, pages 169–182, 1984.

[Pom96] Carl Pomerance. A Tale of Two Sieves. *Notices of the American Mathematical Society*, 43:1473–1485, 1996.

[Pom08] Carl Pomerance. Smooth numbers and the quadratic sieve. *Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography, MSRI Publications*, 44:69–81, 2008.

[Wie86] Douglas H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory*, 32(1):54–62, 1986.