COMPUTING THE RAMANUJAN TAU FUNCTION

DENIS XAVIER CHARLES

ABSTRACT. We show that the Ramanujan Tau function $\tau(n)$ can be computed by a randomized algorithm that runs in time $O(n^{\frac{1}{2}+\epsilon})$ for every $\epsilon > 0$ under GRH. The same method also yields a deterministic algorithm that runs in time $O(n^{\frac{3}{4}+\epsilon})$ for every $\epsilon > 0$ to compute $\tau(n)$ without any assumptions. Previous algorithms to compute $\tau(n)$ require $\Omega(n)$ time.

1. Introduction

Let $\tau(n)$ be the coefficient of q^n in the formal expansion $q\prod_{1\leq n}(1-q^n)^{24} = \sum_{1\leq n}\tau(n)q^n$. The following properties of the τ-function are well known:

(1) If $n, m \in \mathbb{Z}_{>0}$ such that $\gcd(n, m) = 1$ then $\tau(nm) = \tau(n)\tau(m)$.
(2) If $r \geq 1$ and p is a prime then $\tau(p^{r+1}) = \tau(p)\tau(p^r) - p^{11}\tau(p^{r-1})$.

Thus $\tau(n)$ is completely determined by $\tau(p)$ for primes $p|n$. Here is a table of $\tau(p)$ for small prime numbers p.

p	2	3	5	7	11	13
$\tau(p)$	-24	252	4830	-16744	534612	-577738

The importance of the τ-function comes from the fact that it gives the fourier coefficients of a modular form. Namely, the function $\Delta(z) = q\prod_{1\leq n}(1-q^n)^{24}$ where $q = e^{2\pi iz}$ is a cusp form of weight 12 for the full modular group (see [La76]). A famous conjecture of D. H. Lehmer says that $\tau(n)$ is never zero. This conjecture has been verified for all $n \leq 22689242781695999$ [JorKe99]. The function $\tau(n)$ seems to be a hard function to compute. Methods to compute $\tau(n)$ based on recurrence relations that it satisfies or its relations to other arithmetic functions such as $\sigma_k(n)$ require $\Omega(n)$ time steps. Since the number n requires $\log_2 n$ bits these algorithms require exponential time in the length of the input. In this article we show that $\tau(n)$ can be computed in time $O(n^{\frac{1}{2}+\epsilon})$ by a randomized algorithm for every $\epsilon > 0$. Though this algorithm is still an exponential time algorithm it is significantly faster than the other methods. Moreover, algorithms based on recurrences compute values of $\tau(m)$ for $m < n$ when computing $\tau(n)$. Our algorithm has the feature that it does not compute any of the previous values of the τ-function. On the other hand, this algorithm is not well suited to building a table of $\tau(m)$ for all $m < n$ since the table can be built in roughly $O(n)$ time by the other methods, whereas this method would require $O(n^{\frac{3}{2}+\epsilon})$ time. Our algorithm is more suited to computing "spot" values of $\tau(n)$. In the next section we will give the details of the algorithm and prove its running time.

2. The Algorithm

Since we can compute $\tau(n)$ in $O(\log^3 n)$ time provided we know the factorization of the integer n *and* the values of $\tau(p)$ for primes $p|n$, we will concentrate on computing $\tau(p)$ for primes p. There are deterministic algorithms that can factor n in $O(n^{\frac{1}{4}+\epsilon})$ time ([Co93]). We use such an algorithm to find the primes $p|n$. The main idea of the algorithm is to make use of the Selberg Trace formula to compute $\tau(p)$.

Date: March 2003.
Research supported in part by NSF grant CCR-9988202.

Theorem 2.1. [Sel56] *Let $k \geq 4$ be an even integer and let m be an integer > 0. Then the trace of the Hecke operator $T(m)$ on the space of cusp forms $S_k(\Gamma)$ is given by*

$$\mathrm{Tr}\ T(m) = -\frac{1}{2} \sum_{-\infty < t < \infty} P_k(t.m) H(4m - t^2) - \frac{1}{2} \sum_{dd' = m} \min\{d, d'\}^{k-1}.$$

In the above sum $H(D)$ refers to the Hurwitz class number of D, and $P_k(t, N) = \frac{\rho^{k-1} - \overline{\rho}^{k-1}}{\rho - \overline{\rho}}$ where ρ is a complex number satisfying $\rho + \overline{\rho} = t$ and $\rho\overline{\rho} = N$.

Note that the sum is actually finite since $H(D) = 0$ if $D < 0$ and so if $t > 2\sqrt{m}$, $H(4m - t^2) = 0$.

In our case $\Delta \in S_{12}(\Gamma)$ and it is a one dimensional vector space. The Hecke operators are a family of linear operators $T(n) : S_k(\Gamma) \to S_k(\Gamma)$ for $n \geq 1$ an integer. Since $\dim S_{12}(\Gamma) = 1$, Δ is a simultaneous eigenform for every $T(n)$. It is known (see [La76]) that $T(n)\Delta(z) = \tau(n)\Delta(z)$ where $\Delta(z) \in S_{12}(\Gamma)$ is the function defined earlier. Thus the eigenvalue of the n-th Hecke operator is $\tau(n)$. Since $\dim S_{12}(\Gamma) = 1$, we have $\mathrm{Tr}\ T(n) = \tau(n)$ and specializing Theorem 2.1 to our case we get the following result:

Theorem 2.2. *Let p be a prime. Then*

$$\tau(p) = - \sum_{0 < t \leq \sqrt{4p}} P(t, p) H(4p - t^2) \ + \frac{1}{2} p^5 H(4p) \ - 1$$

where

$$P(t, p) = t^{10} - 9t^8 p + 28 t^6 p^2 - 35 t^4 p^3 + 15 t^2 p^4 - p^5$$

and $H(D)$ is the Hurwitz class number.

We will use the above theorem to compute $\tau(p)$. In fact, we only need to show how the Hurwitz class numbers can be computed, since it is easy to compute the above sum. For this task we need the following lemma (see [Co93] Lemma 5.3.7):

Lemma 2.3. *Let $w(-3) = 3, w(-4) = 2$ and $w(D) = 1$ for $D < -4$, and set $h'(D) = \frac{h(D)}{w(D)}$, where $h(D)$ is defined to be the class number of the order of discriminant D in $\mathbb{Q}(\sqrt{D})$ if $D \equiv 0, 1 \mod 4$ otherwise we define $h(D)$ to be zero. Then for $N > 0$ we have*

$$H(N) = \sum_{d^2 | N} h'\left(-\frac{N}{d^2}\right).$$

There are randomized sub-exponential time algorithms to compute the class number (see [Co93]).

Theorem 2.4. *The class number $h(D)$ can be computed deterministically in time $|D|^{\frac{1}{4}+\epsilon}$ for every $\epsilon > 0$, or by a randomized algorithm with expected running time $e^{O(\sqrt{\ln |D| \ln \ln |D|})}$.*

Proposition 2.5. *The Hurwitz class number $H(N)$ can be computed by a deterministic algorithm in time $O(N^{\frac{1}{4}+\epsilon})$ or a randomized algorithm with an expected running time $O(N^{\epsilon})$ for every $\epsilon > 0$.*

Proof : By Lemma 2.3 we have

$$H(N) = \sum_{d^2 | N} h'\left(-\frac{N}{d^2}\right).$$

By Theorem 2.4, the function $h'(D)$ can be computed in time $O(|D|^{\epsilon})$ if we use the randomized algorithm or in time $O(|D|^{\frac{1}{4}+\epsilon})$ if we use the deterministic algorithm. The number of terms in the sum is at most the number of divisors of N. It is known (see [Ten95]) that the number of divisors $d(N) \ll_{\epsilon} N^{\epsilon}$ for every $\epsilon > 0$. Thus the sum can be evaluated by computing each of the terms in the stated time bound. □

Thus putting all these results together we get the following:

Theorem 2.6. *There is a randomized algorithm to compute* $\tau(p)$ *with expected running time* $O(p^{\frac{1}{2}+\epsilon})$ *for every* $\epsilon > 0$.

Theorem 2.7. *There is a deterministic algorithm to compute* $\tau(p)$ *in time* $O(p^{\frac{3}{4}+\epsilon})$ *for every* $\epsilon > 0$.

Acknowledgements: The author would like to thank professor Ken Ono for suggesting the use of the trace formula, and professor Eric Bach for very useful discussions. Thanks are due to Victor Miller who pointed out an error in the statement of Lemma 2.3.

References

[Co93] Cohen, Henri; *Computational Algebraic Number Theory*, Graduate Texts in Math. Vol. 138, Springer-Verlag, 1993.

[JorKe99] Jordan, B.; Kelly, B. III; *The vanishing of the Ramanujan Tau function*, Preprint, 1999.

[La76] Lang, Serge; *Introduction to Modular Forms*, Grundlehren der mathematischen Wissenschaften, Vol. 222, Springer-Verlag, 1976.

[Sel56] Selberg, Alte; *Harmonic analysis and discontinuous groups in weakly symmetric Riemannian spaces with applications to Dirichlet series*, J. Indian Math. Soc., **20**, 47-87, 1956.

[Ten95] Tenenbaum, Gérald; *Introduction to analytic and probabilistic number theory*, Cambridge Studies in Advanced Mathematics, **46**, Cambridge University Press, 1995.

Department of Computer Science, University of Wisconsin-Madison, Madison WI - 53706.
E-mail address: cdx@cs.wisc.edu