Verifying NO instances of 3-SUM in time roughly $n^{3/2}$

Huck Bennett*

January 22, 2026

In this note, we present the beautiful algorithm from work of Carmosino, Gao, Impagliazzo, Mihajlin, Paturi, and Schneider [CGI+16] for showing that 3-SUM is contained in $\mathsf{coNTIME}[n^{3/2} \cdot \text{poly}(\log(n))]$. We will use the version of 3-SUM in which the input is an array $A = [a_1, \ldots, a_n]$ of polynomially bounded integers, and the goal is to decide whether there exist $i, j, k \in [n]$ such that $a_i + a_j + a_k = 0$. So, the algorithm's goal is to certify that a given 3-SUM instance is a NO instance, i.e., that there are no triples $i, j, k \in [n]$ such that $a_i + a_j + a_k = 0$.

The algorithm (given in Theorem 1) uses two key observations about 3-SUM modulo a prime p: (1) that there is a relatively small prime p such that the set R_p of triples (i, j, k) with $a_i + a_j + a_k \equiv 0 \pmod p$ is relatively small, and (2) that it is possible to efficiently *count* the number of 3-SUM solutions modulo a small prime p (i.e., to compute $|R_p|$) using the fast Fourier transform. So, the algorithm takes in a certificate (p, R_p), verifies that the certificate is valid (i.e., that p is in fact a small prime, that R_p is a small set, and that R_p contains all 3-SUM solutions modulo p), and then checks that R_p consists entirely of *false positives modulo* p (i.e., triples that are solutions modulo p, but not solutions over the integers).

Motivation for studying this algorithm comes from the Nondeterministic Strong Exponential Time Hypothesis (NSETH), also introduced in [CGI+16]. Informally, NSETH states that there are no nontrivial nondeterministic algorithms for certifying that instances of k-SAT are unsatisfiable when k is large. That is, NSETH asserts that any such algorithm must take roughly 2^n time, which is how long k-SAT takes to solve deterministically by brute force. On the other hand, the algorithm in Theorem 1 *does* give a nontrivial nondeterministic algorithm for certifying that instances of 3-SUM are NO instances. The fastest known deterministic algorithms for 3-SUM run in roughly n^2 time (up to sub-polynomial factors), and the algorithm in Theorem 1 runs in roughly $n^{3/2}$ time. [CGI+16] notes that this algorithm therefore rules out fine-grained reductions from k-SAT to 3-SUM, assuming NSETH.

Formally, we prove the following theorem.

Theorem 1 ([CGI+16])**.** *There is an $\widetilde{O}(n^{3/2})$-time algorithm that takes as input an array $A = [a_1, \ldots, a_n]$ of n numbers with $a_i \in [-n^c, n^c]$ for some constant $c > 0$ and a certificate of length at most $\widetilde{O}(n^{3/2})$ with the following property.*[1] *If there is no triple of indices $(i, j, k) \in [n]^3$ such that $a_i + a_j + a_k = 0$ then there exists a certificate such that the algorithm accepts, and otherwise the algorithm rejects on all certificates.*

Proof. For an integer $p \geq 2$, let

$$R_p := \{(i, j, k) \in [n]^3 : a_i + a_j + a_k \equiv 0 \pmod p\} \ .$$

A valid certificate consists of a pair (p, R_p) for a prime number p such that $p \leq \widetilde{O}(n^{3/2})$ and $|R_p| \leq \widetilde{O}(n^{3/2})$.

We first prove that such a certificate exists. Define

$$R := \bigcup_p \{(p, (i, j, k)) : (i, j, k) \in R_p\} \ ,$$

where the union is over all prime numbers p. We claim that $|R| = O(n^3 \log n)$. Indeed, each sum $a_i + a_j + a_k$ for $a_i, a_j, a_k \in A$ has magnitude at most $3n^c$, and therefore $|a_i + a_j + a_k|$ has at most $\log_2(3n^c) \leq O(\log n)$

*University of Colorado Boulder. huck.bennett@colorado.edu.

[1] The notation $\widetilde{O}(\cdot)$ suppresses polylogarithmic factors in the argument. That is, $\widetilde{O}(f(n)) := f(n) \cdot \text{poly}(\log(f(n)))$.

many prime factors. So, each triple (i, j, k) is contained in at most $O(\log n)$ sets R_p. The claim then follows since there are n^3 triples (i, j, k).

By an averaging argument, there must exist a prime number p among the first $\lceil n^{3/2} \rceil$ prime numbers such that $|R_p| \leq |R|/n^{3/2} \leq \widetilde{O}(n^{3/2})$. Furthermore, by the prime number theorem, there are $\lceil n^{3/2} \rceil$ prime numbers of magnitude at most $O(n^{3/2} \log n)$, and so $p \leq O(n^{3/2} \log n)$.

We next show how to use the certificate (p, R_p) to certify that no triple of indices (i, j, k) is such that $a_i + a_j + a_k = 0$. The verification algorithm performs three checks, and accepts if and only if they all succeed. First, it checks that p is prime. Second, it checks that $a_i + a_j + a_k \equiv 0 \pmod{p}$ and $a_i + a_j + a_k \neq 0$ for all $(i, j, k) \in R_p$. Third, the algorithm defines the polynomial $q(x) := \sum_{a \in A} x^{a \bmod p}$, and uses the fast Fourier transform to compute $q(x)^3$, which is equal to $\sum_{j=0}^{3(p-1)} b_j x^j$ for some integer coefficients $b_j \geq 0$. It then checks that $b_0 + b_p + b_{2p} = |R_p|$.

Correctness of the algorithm follows by noting that the checks ensure that all 3-SUM solutions modulo p are included in R_p, and that none of these are solutions over the integers. Indeed, $b_0 + b_p + b_{2p}$ is exactly the number of 3-SUM solutions modulo p, and all 3-SUM solutions over the integers are solutions modulo p.

Finally, we analyze the algorithm's running time. Verifying that p is prime using (say) trial division takes $\widetilde{O}(\sqrt{p}) \leq \widetilde{O}(n^{3/4})$ time, checking that each triple (i, j, k) is a solution modulo p but not over the integers takes $|R_p| \cdot \text{poly}(\log n) \leq \widetilde{O}(n^{3/2})$ time, and computing $q(x)^3$ using the fast Fourier transform takes $p \log p \cdot \text{poly}(\log n) \leq \widetilde{O}(n^{3/2})$ time. The theorem follows. □

References

[CGI+16] Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic Extensions of the Strong Exponential Time Hypothesis and Consequences for Non-reducibility. In *ITCS*, 2016. 1