White Paper
An Algorithm: Revisited Goldbach Conjecture

by

Shailesh Kadakia

July 25, 2025

Abstract

In the history of mathematics, Goldbach Conjecture is the most famous and yet it is the most difficult problem in hands of number theory experts. For strong version, it states that every even natural number greater than two can be expressed as the sum of two prime numbers. The weak version of the same predicts that every odd number greater than five is sum of three prime numbers. The strong version has been validated for even numbers as high as 4 x 10^18. However, any comprehensive proof that the conjecture holds good for arbitrarily large even number does not exist. Here, we explore reasons why the task of proving the conjecture if not impossible, it is the most daunting challenge. Moreover, we present a scheme that allows to determine two primes whose sum is equal to chosen even number for any sufficiently large size even number.

Keywords: Prime number, Binary Goldbach Conjecture, Prime Number Theorem, Composite Numbers, Number Theory, Gaps between Consecutive Primes.

An Algorithm: Revisited Goldbach Conjecture

Shailesh Kadakia
July 25, 2025

Abstract

In the history of mathematics, Goldbach Conjecture is the most famous and yet it is the most difficult problem that can be solved by number theory experts. For strong version, it states that every even natural number greater than two can be expressed as the sum of two prime numbers. The weak version of the same predicts that every odd number greater than five is sum of three prime numbers. The strong version has been validated for even numbers as high as 4 x 10^18. However, any comprehensive proof that the conjecture holds good for arbitrarily large even number does not exist. Here, we explore reasons why the task of proving the conjecture if not impossible, it is the most daunting challenge. Moreover, we present a scheme that allows to determine two primes whose sum is equal to chosen even number for any sufficiently large size even number.

1. Introduction

We will begin with a historical note about Goldbach conjectures. Weak conjecture was conceived by Rene Descartes' before Christian Goldbach realized in 1742. But Descartes' did not receive credit during his life-time. He was credited in 1908 when his work was published in edition of *Opuscula Posthuma*. However, Christan Goldbach, a Prussian Mathematicians with his good fortune, was recognized for discovering both strong and weak conjecture in 1742.

Let us understand why proof of this seemingly simple conjecture turned out to be very difficult task. There are two main reasons: a. Gaps between successive primes increases when even numbers under consideration are very large. b. Also, prime count decreases between even numbers in range of consecutive multiples of powers of ten. In Table 1, we have summarized Prime count and average gap vs. numbers size upto 10^25, starting from 1. Also, in Figure 1and 2, we have graphed data prime counting function $\pi(x) = x / \log(x)$ [1] in log scale vs log x and average gap size vs log x. From the figures, it is clear that prime count increases with x but rate of increase decreases exponentially. However, average gap between consecutively higher value of primes increases linearly with x. The first reason implies that there are several even numbers between primes of very large size such as Mersenne Prime number [2] [3] M136279841 = 2^(136,279,841) – 1 and next higher prime number which is not discovered yet. This fact makes it harder to find two prime numbers that are sum of even number greater than M136279841. The second reason results increase of probability that decrease in prime count will cause vanishing (zero prime count increase) of primes at some values of x sufficiently high power of ten, as a limit case. Our reason is supported by evidence that maximum gap between successive prime numbers rises to as high as twenty-five times the average gap value for x number equals 10^{20}. In Figure 3, we have plotted prime size x in powers of ten vs. maximum gap. The effects of these two reasons are a generalized proof of Goldbach Conjecture is not accepted by mathematics community. Therefore, we are presenting a scheme that ensures that even number of any size can be expressed as sum of two prime numbers.

2. Preliminaries

The Goldbach conjecture theoretically remains unproven despite exhaustive efforts to date [4]. Therefore, we are providing a simplified method that could lead us to a generalized proof of the conjecture. In order to achieve our objective, we shall utilize basic properties of integers and prime numbers. A common intuition suggest that any even number is a sum of two odd numbers. Further, based on pattern of prime numbers we can establish a fact that all prime numbers other than two are odd numbers. Moreover, we will prove a unique property of all prime numbers. They are prime if all prime numbers smaller than square root of the prime can not divide the prime number exactly. We shall denote this proof as Prime Test Theorem (PTT).

Theorem 1. Every even integer E is equal to sum of two odd integers A and B.

Proof; Every even number is divisible by two. This implies that $E \div 2 = N$, $N > 0$ an integer

Therefore, $E = 2N$, where N is an even or odd number

Any odd number equals an even integer plus 1

Therefore, $A = 2P + 1$ and $B = 2Q + 1$, here P and Q are positive integers greater than 0

Adding $A + B = 2P + 1 + 2Q + 1$ we get

$= 2P + 2Q + 2$

$= 2 (P + Q + 1)$. Let $N = (P + Q + 1)$

$= 2N$.

We proved the theorem; any even number equals sum of two odd numbers.

Theorem 2. P is a prime number if all prime numbers smaller than $\sqrt{P}$ can not divide P exactly.

Proof: We will prove above theorem by a geometrical technique. An integer $P > 1$ is prime if the number represents area of a rectangle with sides 1 and the number P itself only. We will refer it as a prime rectangle. For integers/rectangles with area other than Prime rectangle areas are called Composite area/rectangles.

Case 1: If P is such that $\sqrt{P}$ is integer than P is composite number because $\sqrt{p}$ is a repeat factor of P.

Case 2: If P is such that $\sqrt{P}$ is a mixed number than P may be composite representing area of a composite rectangle with sides M and N. Thus $P = M \times N$, here $M > 1$ and $N > 1$ are integers. Further, if $M < \sqrt{P}$ implies that $N > \sqrt{P}$. Moreover, $P = M \times N$ implies that $P \bmod (M) = 0$ and $P \bmod (N) = 0$. This contradicts premise M and N can not divide exactly into P. From geometrical stand point there can be only one prime rectangle. Let us define $M = \{P_i \ldots i = 3, \ldots n\}$, $P_n < \lfloor\sqrt{P}$. Therefore, for every M in set of prime numbers $P_i < \sqrt{P}$ if $P \bmod (P_i) \neq 0$ then P must be prime. Thus, we have proved Prime Test Theorem.

Corollary 1: All prime numbers greater than five ends with digits 1, 3, 7, and 9 in the least significant place. We can arrive at this conclusion because all prime numbers are odd. The smallest prime number two is a factor of all other even numbers. Also, number five is a factor of numbers ending with a five or a zero.
Corollary 2: All prime numbers composed of second and higher powers of every prime number are composite numbers.
Corollary 3: Any two prime numbers ending with digits 1, 3, 7, and 9 and single prime number 5 are sufficient to produce all even numbers greater than two ending with digits 2, 4, 6, 8, and 0. Number of ways an even number created is directly correlated to the number of ways two or more odd numbers ending in digits 1, 3, 7, and 9, and number 5 are added to get resultant even number.

With these theorems and corollaries in mind, we will describe an algorithm and steps that will help us achieve our goal of proving the Goldbach Conjecture next. Also, it is established that M136279841 is the largest known Prime number. Therefore, we can assume that all the prime numbers lower than prime number M136279841 have been found and verified by comparison of number test.

3. Candidate Pairs

Elementary subtraction identity of integers allows us to decompose any even number of arbitrarily large size into two odd numbers. Therefore, we will create pairs of odd numbers that will add to produce given starting even number 2N. We shall call them as candidate pairs odd numbers Q1 and Q2. In order to facilitate computation, we will list them in a specific order in which Q1 starts from the lowest value and Q2 starts from the highest value.

Odd Number Pairs: Serial No.	Q1		Q2
1	1		2N – 1
2	3		2N – 3
3	5		2N – 5
4	7		2N – 7
5	9		2N – 9
6	11		2N – 11
7	13		2N - 13
8	●	●	●
9	●	●	●
10	●	●	●
	●	●	●
Last pair	2N – K		2N – K

K is odd and there are N ÷ 2 pairs in all.

In order to improve efficiency of computation, candidates are carefully selected such that the smallest number of pairs are tested for primality. Next, we will discuss steps that will enable us to discover a prime pair from a set of prime pairs. The prime pair is identified by performing square root test of primality. Also, the first identified prime pair found is such that one of the prime numbers is the largest possible prime, when added to the paired prime number that should result into a sum value, starting number 2N.

4. Computational Procedure

We will describe computer algorithm and procedure in great detail here because finding a prime number that satisfies the desired criteria is a NP complete problem. First step in computer program is to screen prime pairs from comprehensive pairs of odd number pairs for a specific starting even number 2N. The step ensures that we will perform square root tests only on the eligible candidates which are numbers ending in 1, 3, 7, and 9 and which are not perfect powers of any prime number. The square root test is not performed on prime complements corresponding to number 2 because adding 2 to an odd prime will not result into sum 2N. However, the square root test is executed for testing a prime pair 5 and 2N – 5 along with prime pairs ending in 1, 3, 7, and 9. After a set of qualified prime pairs is created, square root prime test is performed on Q2 side of the pair first. When a prime number on Q2 side is found the square root test is performed for a corresponding prime number on Q1 side. If the number on Q1 side is Prime we have confirmed that both Q1 and Q2 numbers identified are primes. In case number on Q1 side is not prime, we should find next set of prime numbers on Q2 side and Q1 side by repeating the same steps. At this point our search for prime numbers should be complete. Next, we will describe specific steps in more detail. A flowchart of a Python program is displayed in Figure 4.

Step 1. Generate Candidate

Development of computer programs that test primality is a very sophisticated and expensive process. It is well known that modern computers are the most powerful and are of the highest speed suitable for this application. However, performing prime test on numbers of size M136279841 requires huge amount of memory array because the number is composed of millions of decimal digits. It is impossible to perform square root test on this size of number manually. Therefore, program code is written in languages such as Python that can implement very efficient data structures and can utilize features of object-oriented programming to optimize resource usage. One of the most useful features of computer program is efforts spent in its development can be utilized repeatedly. First, step in this program is to allocate memory to store an array of predetermined size of numbers. Then, the array is initialized to prime number of increasing sizes in Q1 array and decreasing sizes in Q2 array. The values of members in Q1 and Q2 arrays are initialized as follows.

Prime Number Pairs:	Serial No.	Q1		Q2
	1	3		2N – 3
	2	5		2N – 5
	3	7		2N – 7
	4	11		2N – 11
	5	13		2N – 13
	6	●	●	●
	7	●	●	●
	8	●	●	●
		●	●	●
	Last square toot test pair	P		2N – P

P+1	●	●	●
P+2	●	●	●
P+3	●	●	●
	●	●	●
P+K	K		1

K is odd and there are N ÷ 2 pairs in all. Also, $P = \sqrt{2N}$.

The rows exclude perfect powers of all prime numbers. Next, segment of program searches for first prime number in Q2 array by performing square root test.

Step 2. Test Primality

The Section of program loop which performs square root test is capable of testing a member in array either Q2 or Q1 a prime or composite is called primality loop. The code performs modulo division operation sequentially dividing the Q2 member by primes in set Q1 array 3, 5, 7, … till $P = \sqrt{2N}$ and checks for zero remainder. If remainder is zero in any division step than the member is composite. After that next lower number in array Q2 is tested for primality. When a prime number is found in Q2 array. The square root test is performed on corresponding prime number in Q1 array to make sure that the prime pair in both Q1 and Q2 arrays are primes.

The square root test plays a crucial role in primality test and is a very important test in proof of the Goldbach conjecture. Therefore, we should examine every edge case in the test with regards primality of numbers and critically analyze results of the test. For this purpose, we will calculate upper bound on number of square root tests performed in Q2 odd array. Total number of odd pairs in Q1 and Q2 array for any even number 2N is N. However, prime number only ends with 1, 3, 7, and 9. Therefore only 4 out of five odd numbers are tested. Further, from the highest element in Q2 array we exclude square root test for 2N – 1 and 2N – 9 because 1 is neither prime nor composite and number 9 is composite. Thus, upper bound on square root test performed in elements of Q2 array is computed as follows.

(4/5) ● (N – 1) pairs of odd numbers ending with 1, 3, 7, and 9 for prime numbers greater than 10

\- 2 subtract for 2N – 1 and 2N - 9 because 9 is composite and 1 is not prime

\+ 3 add tests for 2N – 3, 2N – 5, and 2N – 7.

However, in reality we have to perform tests loop are performed for P = floor $\lfloor \sqrt{2N}$ times. Therefore, we will replace (N -1) by P - 1. The actual test loop count will be (4/5) ● (P – 1) + 1 – (all the powers of prime times contained in 2N). The most interesting edge case the number $2N = Q^2 + 1$, here Q is the largest prime number known to date such as Luke Durant prime number M136279841. For our discussion here, we will designate it as number M. This is the number that can be tested for primality using the square root test. According to Goldbach conjecture the value of this even number can be expressed as the sum of two prime numbers contained in (4/5) ● (P – 1) + 1 row of odd numbers in Q1 and Q2 arrays. Therefore, upper bound for the even number obeying Goldbach conjecture which are tested and verified is M + 1. If we account composition of both Prime and Composite sum of any two elements, one from each array Q1 and Q2, theoretically we can get result of 2M. Therefore, upper bound of a pair of numbers tested for Goldbach conjecture verification by applying square root test is 2M.

Step 3. Validation

In this section, we will investigate limitations of our algorithm to identify a prime number whose value is higher than the largest known prime. The largest known prime number is 2^(136,279,841) – 1, also known as M136279841. It has 41,024,320 digits and was discovered in October 2024 by Luke Durant, a contributor to the Great Internet Mersenne Prime Search (GIMPS) [3]. One problem, we run into while applying square root algorithm is what if the starting even number 2N is greater than square of M136279841. In that case we are attempting to discover a prime that was not discovered before the program is ran. A question is How do we know that a prime number exists one that is greater than M136279841. Therefore, we will prove that many prime numbers exist which are larger than Luke Durant number. Our proof is based on a popular and a very good approximation prime counting function $\pi(x) = x/[\ln(x)]$. The function counts number of prime numbers lower than number x.

For our proof, we will use a slightly modified version of the prime counting function. We are interested in counting prime numbers lower than a number which is a whole power of number 10. Also, we will modify notation of $\pi(x)$ to $\pi(n)$, where n is the exponent of the power of 10. Further, we will change base of logarithm in denominator from e to 10. Therefore, updated prime counting function for a number 10^n is

Prime counter $\pi(n) = 10^n /[(\log 10^n) / \log e] = 0.4343\ 10^n / (\log 10^n)$

For a number $10^{(n+1)}$, $\pi(n + 1) = 0.4343\ 10^{n+1} / (\log 10^{n+1})$ and

Also, for number 10^n, $\pi(n) = 0.4343\ 10^n / (\log 10^n)$ Let us take ratio $\pi(n + 1) / \pi(n)$

$$\pi(n + 1) / \pi(n) = 0.4343\ 10^{n+1} / (\log 10^{n+1}) \div 0.4343\ 10^n / (\log 10^n)$$

$$= 10 \bullet 10^n / (n + 1) \log 10 \div 10^n / n \log 10$$

$$= 10\, n/(n + 1) = 10/(1 + 1/n)$$

We wish to find the value of this ratio as n approaches ∞. Therefore, we will take limit of above expression

$$\lim_{n \to \infty} [\pi(n + 1) / \pi(n)] = \lim_{n \to \infty} 10/(1 + 1/n) = 10 > 1.$$

Thus, we have proved that ratio of prime count between a number of next larger power of 10 to previous power of ten is 10. The number of prime counts increases ten-fold for each ten-fold increase in size of a number. In essence our square root test is guaranteed to find higher prime number because they always exist. We can substantiate our conclusion by emphasizing fact principle of reoccurrence which says that number pattern composed of interleaved Prime/Composite should continue indefinitely as long as the set of numbers remains unchanged. Further, primality of a number is distinguishing feature and is a fundamental property of decimal number system. Therefore, our proof satisfies and meets required generalization criteria. Moreover, examination of prime number pattern reveals a periodicity. The distance between successive prime numbers ending with same digit such as 11 and 31, 31 and 41 is always a multiple of ten. Similarly distances between prime number ending with digit 3 such as 13, 23, 43, 53, 73, 83, 103 is either 10, 20, 30, 40, and so on. Therefore, we stipulate that probability of finding a prime number for an arbitrarily large size number is certain.

Therefore, successful completion of primality test and execution of our algorithm proves that for an arbitrarily large size even number the Goldbach Conjecture holds good. Furthermore, by proving strong conjecture, we, have also proven weak conjectures because an arbitrarily large size odd number can be split into an even number and prime number 3. Therefore, every odd number greater than seventeen can be expressed as the sum of three distinct prime numbers.

5. Computational Considerations

In twenty first Century Computer Systems are the most valuable resource that provide optimum solution to problems of prime number processing complexity. One of the major problems in performing square root tests is in division operation, we encounter operands of enormous size numbers. For instance, it requires forty-one million twenty-four thousand three hundred twenty digits to represent Duran Prime number M136279841 in ten base number system [12]. Therefore, size of memory arrays allocated to perform square root test increases exponentially. Clever programmers should employ run time variable member size arrays allocation scheme and object-oriented programming features to improve throughput of computers and speed.

Second problem is performing division operation on such large size operands. Very few computers hardware and software systems are designed to handle and perform arithmetic operations on this size operands. Further, revision of IEEE floating point standards are required to address needs specific for this application. Moreover, for increasing speed of computation, thousands of processors must concurrently execute programs and exchange results without any errors. Therefore, special programs are developed to confirm results of these tests. Since, it is very difficult to estimate financial tangible gains from this type of project, it is imperative that investment contribution should come from mathematics community globally. We anticipate that this effort will bring people together in sharing their resources for mutual benefit.

6. Concluding Remarks

This white paper presented an algorithm to find two Prime numbers which satisfied famous Goldbach Conjecture. The algorithm partitions odd numbers into prime pairs in a hierarchy and applies square root test recursively to find at least one pair of primes. Further, segmentation of even into odd numbers ensures that primes found add to a sum equals to the starting even number. I hope that this white paper inspires to find even larger size prime numbers because algorithm presented by us should improve search of prime numbers larger than prime number discovered so far. We will appreciate any constructive comments and suggestion from reviewing community. This will help us improve proofs for this conjecture as well as other problems in number theory.

Acknowledgements

I wish to thank my daughter Sheryl Kadakia a math major from University of Texas for critically analyzing my proof and providing constructive comments. Further, I wish to thank my friend, Professor Munawar Karim for taking interest in this proof. I wish to thank Dr. Derek Muller at Veritasium for producing a video, an excellent review of Goldbach conjecture title, “The Obviously True Theorem No One Can Prove”. His YouTube inspired me to work on this and other similar projects. Also, I wish to extend sincere thanks to my friend Nia Stefani, Managing Partner Xnergy Financial and her son West Stefani for supporting this effort. Also, I wish to thank my colleague at Mathnasium Mr. Marcus Stidham for interesting conversations and feedback on my Goldbach Conjecture Project. Last but not the least, I wish to extend my sincere thanks to wikipedia pages and Google Chrome browser to make results available from prior research fellows in Mathematics.

References

[1] Prime counting function wikipedia https://en.wikipedia.org/wiki/Prime-counting_function

[2] Mersenne Prime number https://mathworld.wolfram.com/MersennePrime.html

[3] Great Internet Mersenne Prime Search GIMPS https://www.mersenne.org/primes/

[4] Budee, U. Z. (2024). Discover a Proof of Goldbach's Conjecture

[5] E. Markakis, C. Provatidis, N. Markakis. (2013), An Exploration on Goldbach's Conjecture, International Journal of Pure and Applied Mathematics Volume 84 No. 1 2013, 29-6

[6] Xin Wang, Two Ways to Prove Goldbach Conjecture, London Journal of Research in Science: Natural and Formal, Volume 21 | Issue 6 | Compilation 1.0

[7] Phillippe S. (2024), Proof of the Binary Goldbach Conjecture, University of Pierre et Marie Curie Paris, France

[8] Redha M. B., The Strong Goldbach Conjecture: Proof for All Even Integers Greater than 362

[9] Herald A. H. (2014), The Ternary Goldbach Problem, Proceedings of the International Congress of Mathematicians, Seoul

[10] "The Obviously True Theorem No One Can Prove" You tube link from Veritasium https://www.youtube.com/watch?v=x32Zq-XvID4&t=8s

[11] Prime gap studies wikipedia https://en.wikipedia.org/wiki/Prime_gap

[12] Largest known prime number wikipedia https://en.wikipedia.org/wiki/Largest_known_prime_number

TABLE 1: Number of Primes and average gaps vs Powers of 10.

Power of 10	$\pi(x)$ accurate	Average Gap
10 = 10^1	4	1.25
100 = 10^2	25	4
1000 = 10^3	168	5.95238095
10000 = 10^4	1229	8.13669650
100000 = 10^5	9592	10.42535446
1000000 = 10^6	78498	12.73917807
10000000 = 10^7	664,579	15.04712006
100000000 = 10^8	5,761,455	17.35672673
1000000000 = 10^9	50,847,534	19.66663713
10000000000= 10^{10}	455,052,511	21.97548581
100000000000= 10^{11}	4,118,054,813	24.28330961
1000000000000= 10^{12}	37,607,912,018	26.59014942
10000000000000= 10^{13}	346,065,536,839	28.89626078
100000000000000= 10^{14}	3,204,941,750,802	31.20181513
1000000000000000= 10^{15}	29,844,570,422,669	33.50693228
10000000000000000= 10^{16}	279,238,341,033,925	35.81170108
100000000000000000= 10^{17}	2,623,557,157,654,233	38.11618882
1000000000000000000= 10^{18}	24,739,954,287,740,860	40.42044655
10000000000000000000= 10^{19}	234,057,667,276,344,607	42.72451365
100000000000000000000= 10^{20}	2,220,819,602,560,918,840	45.02842099
1000000000000000000000= 10^{21}	21,127,269,486,018,731,928	47.33219315
10000000000000000000000= 10^{22}	201,467,286,689,315,906,290	49.63584989
100000000000000000000000= 10^{23}	1,925,320,391,606,803,968,923	51.93940730
1000000000000000000000000= 10^{24}	18,435,599,767,349,200,867,866	54.24287859
10000000000000000000000000= 10^{25}	176,846,309,399,143,769,411,680	56.54627475
100000000000000000000000000= 10^{26}	1,699,246,750,872,437,141,327,603	58.88496049
1000000000000000000000000000=10^{27}	16,352,460,426,841,680,446,427,399	61.15287693
10000000000000000000000000000=10^{28}	157,589,269,275,973,410,412,739,598	63.45609720

Source: https://t5k.org/howmany.html#table

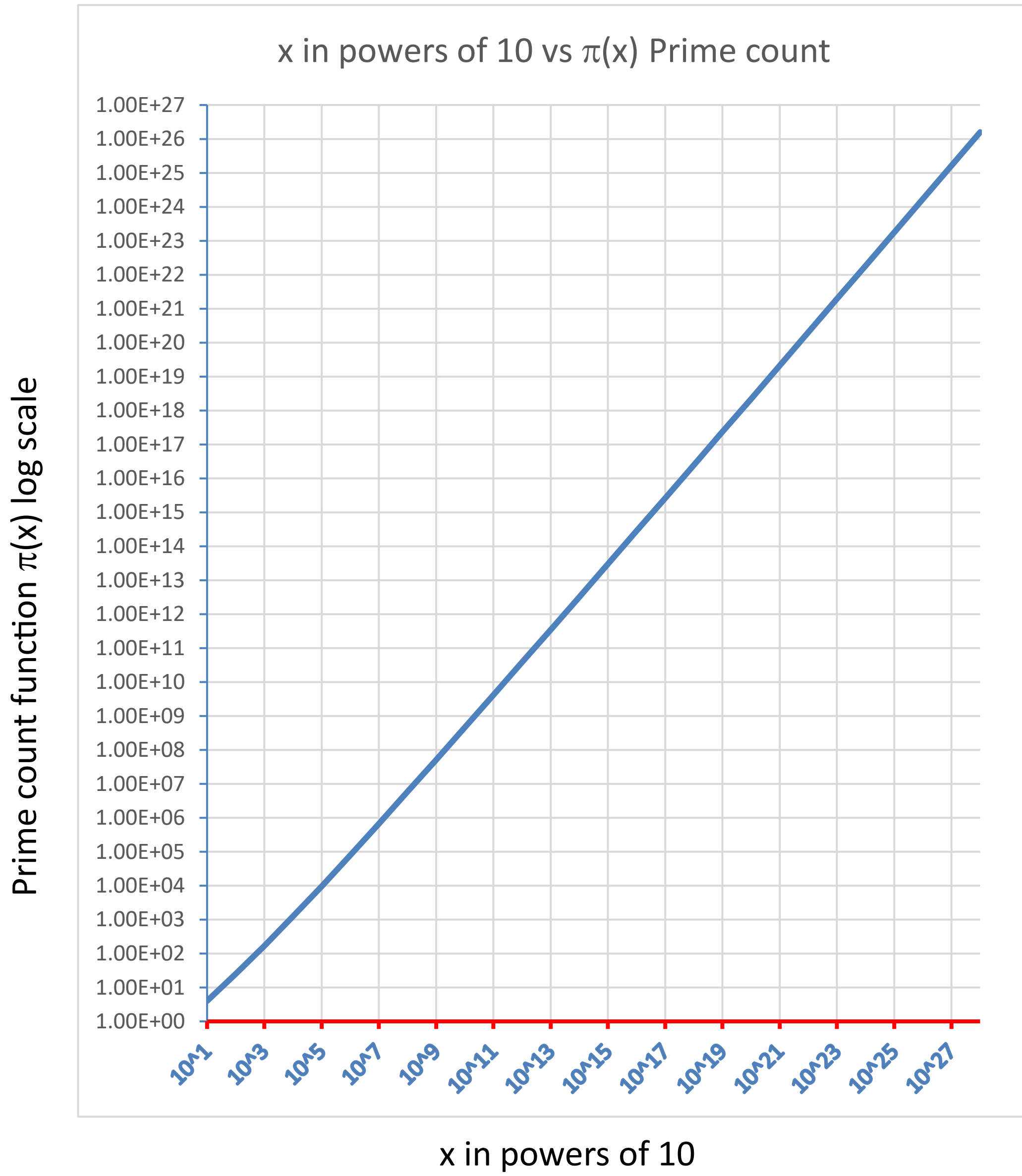

Figure 1: Number powers of ten x (upto 10^{28}) vs. Prime counts $\pi(x)$ on a log scale.

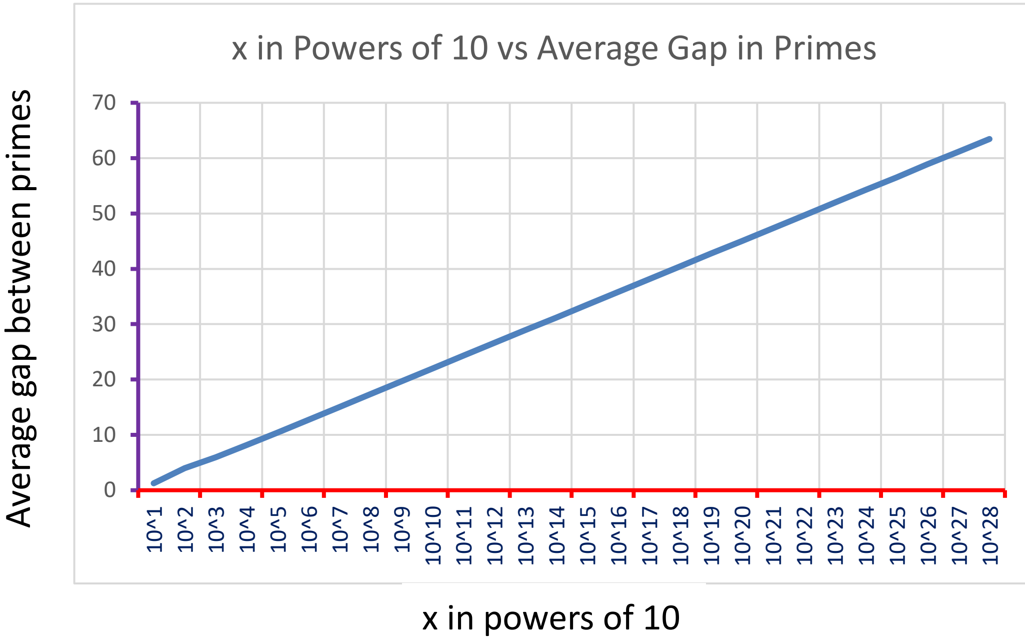

Figure 2: Number powers of ten x (upto 10^{28}) vs. Average gap between Primes.

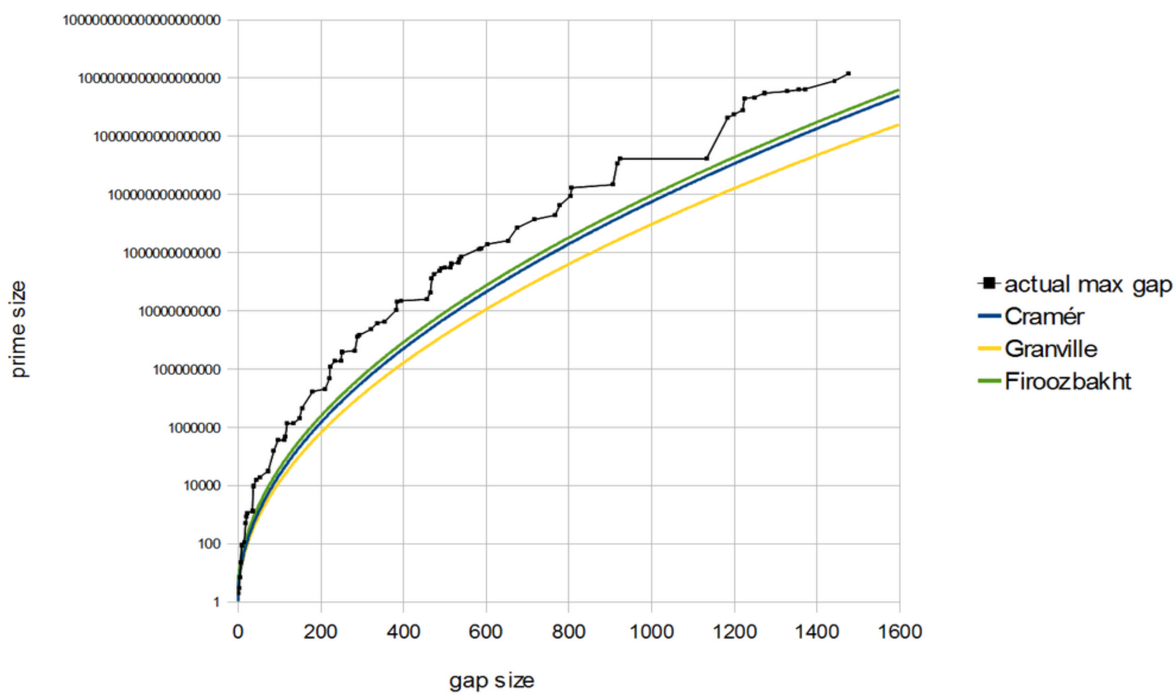

Figure 3: Prime number powers of ten (upto 10^{20}) vs. Maximum gap

Source wikipedia: https://en.wikipedia.org/wiki/Prime_gap

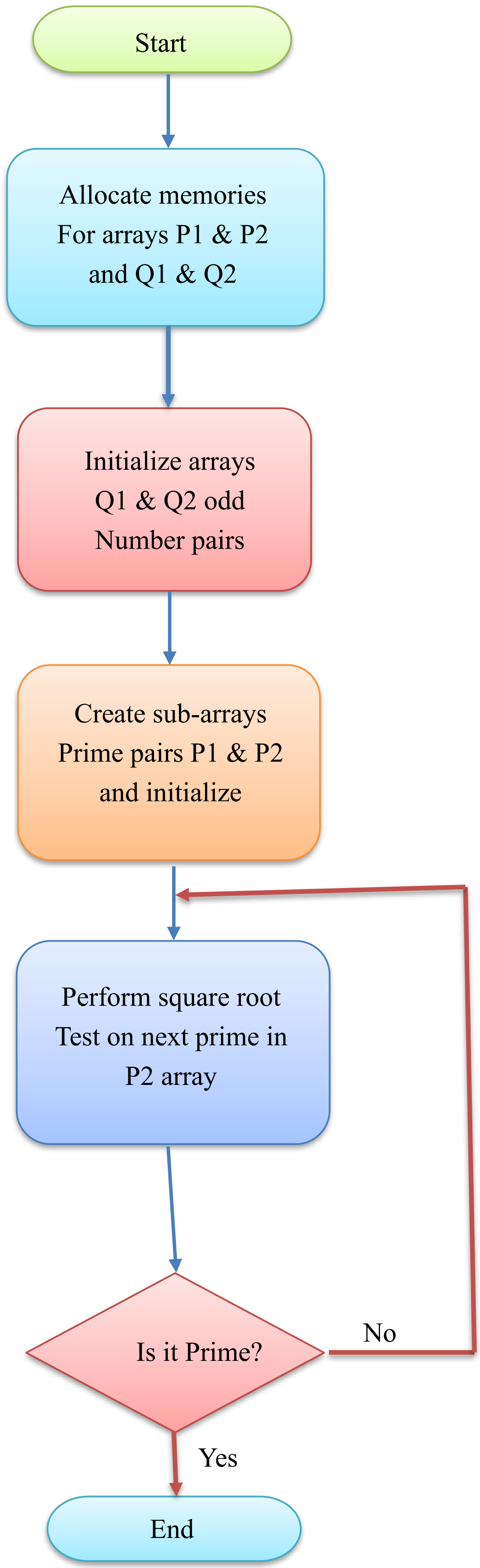

Figure 4: A flowchart that implement prime search algorithm.