Scientific Project

10 Open Problems in Discrete Mathematics & Number Theory

Aleda Islam Mou

Seminar Group - MA19w2-M

Prof. Dr. Klaus Dohmen

Applied Mathematics for Network & Data Sciences

HOCHSHULE MITTWEIDA

Introduction

Number theory and Discrete mathematics are fascinating areas of study that have been the focus of mathematical research for centuries. Despite the many breakthroughs that have been made in these fields, there are still many open problems that continue to challenge mathematicians. These problems are often highly abstract and require a deep understanding of number theory, algebra, geometry, and other branches of mathematics. These problems pose significant challenges to mathematicians and continue to motivate ongoing research. These open problems include questions about the distribution of prime numbers, the existence of perfect numbers, and many others. The resolution of these problems would advance our understanding of number theory and discrete mathematics and have far-reaching implications for other fields such as cryptography, computer science, and physics. In this context, pursuing these open problems remains a fundamental aspect of contemporary mathematical research. This project presents ten open problems in Discrete Mathematics & Number Theory with their corresponding example codes. ‘

Problem 01

The Twin Prime Conjecture:

This conjecture states that there are infinitely many twin primes (pairs of primes that differ by 2). [Dunham (2013)]

Prime number: A prime number is a positive integer greater than 1 that has no positive integer divisors other than 1 and itself. In other words, a prime number is a whole number greater than 1 that can only be divided evenly by 1 and itself.

For example, a few prime numbers are $2, 3, 5, 7, 11$, and 13.

Twin prime: A twin prime is a prime number pair that is either 2 less or 2 more than another prime number. In other words, it is a prime number pair $(p, p+2)$ or $(p-2, p)$ such that both numbers are prime where p is a prime number.

For example, the numbers $(3, 5)$ and $(5, 7)$ are twin primes. The existence of infinite twin primes is still an open problem in number theory.

Sage

The Twin Prime Conjecture: This is a conjecture that states that there are infinitely many twin primes (pairs of primesthat differ by 2).

```
# To check if a number is prime.
def is_prime(number):
    if number <= 1:
        return False
    for value in range(2, int(number ** 0.5) + 1):
        if number % value == 0:
            return False
    return True

# To generate twin primes up to the number n.
def twin_primes(n):
    primes = []
    for number in range(3, n + 1):
        if is_prime(number) and is_prime(number + 2):
            primes.append((number, number + 2))
    return primes

# Example
print(twin_primes(200))
```

```
[(3, 5), (5, 7), (11, 13), (17, 19), (29, 31), (41, 43), (59, 61), (71, 73), (101, 103), (107, 109), (137, 139), (149, 151), (1
79, 181), (191, 193), (197, 199)]
```

Problem 02

Goldbach's Conjecture:

Is every even integer greater than 2 the sum of two prime numbers? [Wang (2002)]

The Goldbach conjecture is a famous unsolved problem in number theory that states that every even integer greater than 2 can be expressed as the sum of two prime numbers.

In other words, for any even number n greater than 2, there exist two prime numbers p and q, such that $n = p + q$. For example, the even number 8 can be expressed as the sum of the prime numbers 3 and 5 (i.e., $8 = 3 + 5$).

Despite numerous attempts to prove the Goldbach conjecture, the complete proof is yet to be found. However, there is strong numerical evidence to support the conjecture, as it has been verified for all even numbers up to $4 * 10^{18}$ through computer calculations.

The Goldbach conjecture has inspired many related problems and conjectures in number theory, and it remains an active area of research. Many mathematicians believe the conjecture is true, and efforts to prove or disprove it are ongoing.

Sage

Goldbach's Conjecture: Is every even integer greater than 2 the sum of two prime numbers?

```
# To check if a number is prime.
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5)+1):
        if n % i == 0:
            return False
    return True

# Compute Goldbach prime pair
def goldbach(n):
    for i in range(2, n):
        if is_prime(i) and is_prime(n-i):
            return (i, n-i)
    return None

# Number to consider
n = 100

print(f"Goldbach pair for {n}: {goldbach(n)}")
```

```
Goldbach pair for 100: (3, 97)
```

Problem 03

Fibonacci square conjecture:

It asks whether there are any perfect squares in the Fibonacci sequence, other than 0 and 1. [Cohn (1967)]

Fibonacci sequence, the sequence of numbers $1, 1, 2, 3, 5, 8, 13, 21, \ldots$, each of which, after the second, is the sum of the two previous numbers; that is, the nth Fibonacci number $F_n = F_{n-1} + F_{n-2}$.

This conjecture was first posed by the French mathematician André N. Dujella in 1998, and it remains an open problem in number theory. Despite extensive computational and theoretical work on this problem, no one has been able to prove or disprove the conjecture.

However, it is known that if there is a perfect square in the Fibonacci sequence other than 0 and 1, then it must occur in the sequence infinitely many times. This is a consequence of the periodicity of the Fibonacci sequence modulo any fixed integer.

In particular, the Fibonacci sequence modulo any prime p has a period of at most $(p^2 - 1)$, which implies that there are only finitely many distinct remainders when the Fibonacci sequence is divided by any fixed prime p.

If there is a perfect square in the Fibonacci sequence, then it must occur in one of these finite sets of remainders modulo p, and hence it must occur infinitely many times in the sequence.

Sage

Fibonacci square conjecture: It asks whether there are any perfect squares in the Fibonacci sequence, other than 0 and 1.

```
# First n Fibonacci numbers
n = 20
fibonacci_numbers = [i for i in fibonacci_sequence(0, n + 1)]
print(fibonacci_numbers)
```

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765]
```

```
fibonacci_numbers = [0, 1]

# Create a fibonacci number and append the list of fibonacci numbers
for i in range(2, 20):
    fibonacci_number = fibonacci_numbers[i-1] + fibonacci_numbers[i-2]
    fibonacci_numbers.append(fibonacci_number)

    # Check if the ith index of the list of fibonacci numbers is a perfect square
    if fibonacci_numbers[i] > 1 and fibonacci_numbers[i] == int(fibonacci_numbers[i]**0.5)**2:
        print(fibonacci_numbers[i])
```

```
144
```

Problem 04

The odd perfect number problem:

The odd perfect number problem is one of the oldest and most famous unsolved problems in mathematics. The problem seeks to determine whether any odd perfect numbers exist. [Dris (2012)]

Perfect number: In number theory, a perfect number is a positive integer that is equal to the sum of its proper divisors (excluding itself). For example, 6 is a perfect number because its proper divisors (excluding 6) are $1, 2$, and 3, and $1 + 2 + 3 = 6$.

Some other examples of perfect numbers are $28, 496$, and $8, 128$. The first few perfect numbers are:

$6 = 1 + 2 + 3$

$28 = 1 + 2 + 4 + 7 + 14$

$496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248$

Perfect numbers have been studied since ancient times, and they have many interesting properties and connections to other areas of mathematics, such as number theory, algebra, and geometry. The study of perfect numbers is an active area of research in modern mathematics.

Sage

The odd perfect number problem: The odd perfect number problem is one of the oldest and most famous unsolved problems in mathematics. The problem seeks to determine whether any odd perfect numbers exist.

```
# Sum of divisors of a number
def sum_divisors(number):
    return sum(value for value in range(1, number) if number % value == 0)

# Check if a number is perfect
def is_perfect(number):
    return number == sum_divisors(number)

# Generate the first n perfect numbers
def perfect_numbers(n):
    i = 1
    result = []
    while len(result) < n:
        if is_perfect(i):
            result.append(i)
        i += 1
    return result

perfect_numbers(3)
```

```
[6, 28, 496]
```

Problem 05

The Beal Conjecture:

The Beal Conjecture is a problem in number theory that is related to Fermat's Last Theorem. The conjecture is named after Andrew Beal, a businessman and number theory enthusiast who proposed it in 1993. [Beal (1997)]

By **Fermat's Last Theorem** no three positive integers a, b, and c that satisfy the equation $a^n + b^n = c^n$ for any integer value of n greater than 2.

The Beal Conjecture is a generalization of Fermat's Last Theorem to equations of the form $ax^p + by^q = cz^r$, where a, b, c, p, q, and r are positive integers greater than 2, and x, y, and z are positive integers greater than 1.

It states that if there are positive integers a, b, c, p, q, r, x, y, and z that satisfy the equation $ax^p + by^q = cz^r$, and if p, q, and r are all greater than 2, then a, b, and c must have a common factor.

The Beal Conjecture has not yet been proven or disproven, despite significant efforts from mathematicians over the years. In 2013, Beal offered a prize of $\$100,000$ for a proof of the conjecture or a counterexample that disproves it.

Sage

The Beal Conjecture: The Beal Conjecture is a generalization of Fermat's Last Theorem to equations of the form ax^p + by^q = cz^r, where a, b, c, p, q, and r are positive integers greater than 2, and x, y, and z are positive integers greater than 1

```
# Define the equation ax^p + by^q = cz^r
a = 2
b = 3
c = 4
p = 3
q = 4
r = 5

# Check if there exist positive integers x, y, and z that satisfy the equation
for x in range(2, 200):
    for y in range(2, 200):
        for z in range(2, 200):
            if a*x**p + b*y**q == c*z**r:
                print("Solution found: x =", x, "y =", y, "z =", z)
                common_factor = gcd(gcd(a,b),c)
                print("The common factor is:", common_factor)
```

```
Solution found: x = 108 y = 36 z = 18
The common factor is: 1
```

Problem 06

The Erdős-Straus conjecture:

The conjecture states that every positive integer n that is 2 or more can be expressed as the sum of at most 4 distinct reciprocals of integers. [Ionascu and Wilson (2010)]

More formally, the conjecture can be stated as follows: For every positive integer n, there exist distinct positive integers x, y, and z such that

$$4/n = 1/x + 1/y + 1/z$$

For example, the number 5 can be expressed as $1/2 + 1/3 + 1/10$, which uses three distinct unit fractions The conjecture is named after the Hungarian mathematician Paul Erdős and the American mathematician Ernst Straus, who independently proposed it in the mid-20th century. The conjecture has been verified for all sufficiently large positive integers, but a general proof remains elusive. Computer searches have verified the truth of the conjecture up to $n \leq 10^{17}$.

Sage

```
# Set prime condition
def prime_condition(a, b, c, n):
    if is_prime(n):
        m = ZZ.random_element(1,10)
        a *= 1/m
        b *= 1/m
        c *= 1/m
        print(f"4/{ m * n} = {a} + {b} + {c}" , f" This statement is: {4/(m * n) == a + b + c}")


# To demonstrate a few cases of Erdös Straus conjecture
def erdos_straus(n):

    # Check for n modulo 4
    if n % 3 == 2:
        a = 1/n
        b = 1/((n + 1)/3)
        c = 1/((n * (n + 1))/3)
        print(f"4/{n} = {a} + {b} + {c}" , f" This statement is: {4/n == a + b + c}")
        prime_condition(a, b, c, n)

    # Check for n modulo 4
    if n % 4 == 3:
        a = 1/((n + 1)/4)
        b = 1/(((n + 1)/2) * n)
        c = 1/(((n + 1)/2) * n)
        print(f"4/{n} = {a} + {b} + {c}" , f" This statement is: {4/n == a + b + c}")
        prime_condition(a, b, c, n)

        if is_even(1/b):
            d = (((n + 1)/2) * n)//2
            b = 1/(d + 1)
            c = 1/(d * (d + 1))
            print(f"4/{n} = {a} + {b} + {c}" , f" This statement is: {4/n == a + b + c}")
            prime_condition(a, b, c, n)

        else:
            d = (((n + 1)/2) * n)//2
            b = 1/(d + 1)
            c = 1/((d + 1) * (2*d + 1))
            print(f"4/{n} = {a} + {b} + {c}" , f" This statement is: {4/n == a + b + c}")
            prime_condition(a, b, c, n)

for n in range(2, 10):
    erdos_straus(n)
```

```
4/2 = 1/2 + 1 + 1/2  This statement is: True
4/10 = 1/10 + 1/5 + 1/10  This statement is: True
4/3 = 1 + 1/6 + 1/6  This statement is: True
4/18 = 1/6 + 1/36 + 1/36  This statement is: True
4/3 = 1 + 1/4 + 1/12  This statement is: True
4/12 = 1/4 + 1/16 + 1/48  This statement is: True
4/5 = 1/5 + 1/2 + 1/10  This statement is: True
4/20 = 1/20 + 1/8 + 1/40  This statement is: True
4/7 = 1/2 + 1/28 + 1/28  This statement is: True
4/56 = 1/16 + 1/224 + 1/224  This statement is: True
4/7 = 1/2 + 1/15 + 1/210  This statement is: True
4/49 = 1/14 + 1/105 + 1/1470  This statement is: True
4/8 = 1/8 + 1/3 + 1/24  This statement is: True
```

Problem 07

The Collatz Conjecture:

This is a conjecture that states that starting from any positive integer and applying the simple rule "if the number is even, divide it by 2; if it is odd, triple it and add 1" will eventually reach 1. [Machado et al. (2021)]

It is a famous unsolved problem in mathematics that deals with iterative sequences generated by a simple rule. The conjecture is named after Lothar Collatz, a German mathematician who first proposed it in 1950.

The conjecture starts with a positive integer n and generates a sequence by applying the following rule repeatedly: if n is even, divide it by 2; if n is odd, multiply it by 3 and add 1. The sequence continues until it reaches the number 1.

For example, starting with the number 6, the sequence generated by the Collatz conjecture is: $6, 3, 10, 5, 16, 8, 4, 2, 1$.

The Collatz conjecture states that for any positive integer n, the sequence generated by the above rule will eventually reach the number 1, no matter what initial value of n is used. The problem is notoriously difficult, and it is not well understood why the sequence behaves in the way that it does.

Sage

The Collatz Conjecture: This is a conjecture that states that starting from any positive integer and applying the simple rule “if the number is even, divide it by 2; if it is odd, triple it and add 1” will eventually reach 1.

```
# Compute Collatz of a number
def collatz(number):
    sequence = [number]
    while number != 1:
        if number % 2 == 0:
            number = number // 2
        else:
            number = 3*number + 1
        sequence.append(number)
    return sequence

collatz(15)
```

```
[15, 46, 23, 70, 35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1]
```

Problem 08

The ABC conjecture:

This conjecture was first proposed by Joseph Oesterlé and David Masser in 1985. It is stated in terms of three positive integers a, b, and c that are relatively prime and satisfy $a + b = c$. [Browkin (2000)]

The conjecture asserts that for any given value of epsilon greater than zero, there are only finitely many triples (a, b, c) that violate a certain inequality involving the prime factors of a, b, and c.

The statement of the ABC conjecture involves the concept of the radical of a positive integer, which is the product of the distinct prime factors of the integer. The conjecture asserts that for any three relatively prime positive integers a,b, and c with $a + b = c$, the inequality

$$c < K_\epsilon * rad(a * b * c)^{1+\epsilon}$$

holds, where $rad(n)$ denotes the product of the distinct prime factors of n and K_ϵ is a constant that depends on ϵ.

Sage

The ABC conjecture: It is a major unsolved problem in number theory which was first proposed by Joseph Oesterlé and David Masser in 1985. It is stated in terms of three positive integers a, b, and c that are relatively prime and satisfy a + b = c.

```
# Compute the radical of a number
def rad(number):
    prime_divisors = prime_factors(number)
    return prod(prime_divisors)

def abc_conjecture(a, b, c, K, epsilon):
    if gcd(a, b) == 1 and gcd(a, c) == 1 and gcd(b, c) == 1:
        if a + b == c:
            if c < K * rad(a * b * c) ** (1 + epsilon):
                return True
    return False

a = 3
b = 5
c = 8
K = 1
epsilon = 0.5

print(abc_conjecture(a, b, c, K, epsilon))
```

```
True
```

Problem 09

The Lonely Runner Conjecture:

The Lonely Runner Conjecture is a problem in mathematics that concerns a group of runners who start running at the same time on a circular track of unit length. Each runner has a different positive integer speed, and they all run in the same direction. [Pandey (2009)]

The conjecture asks whether there is a moment in time when each runner is "lonely", that is, at a distance of at least $1/s$ from any other runner, where s is the runner's speed.

Formally, let n be a positive integer and let $s1, s2, ..., sn$ be positive integers such that $s1 < s2 < ... < sn$. Suppose that n runners start running at the same time on a circular track of unit length. The speed of the ith runner is si. Then, the conjecture states that there exists a moment in time t such that for each i $(1 \leq i \leq n)$, the distance between runner i and the closest other runner j (where $i \neq j$) is at least $1/si$.

In other words, the conjecture asks whether there is a point in time when each runner is at least a certain distance away from all the other runners, where the distance depends on the runner's speed. This distance is proportional to the inverse of the runner's speed, which means that faster runners must be farther away from each other in order to be considered "lonely".

Sage

```
: # To verify the lonely runner conjecture
def lonely_runner_conjecture(s):
    n = len(s)
    for t in range(1, n+1):
        lonely = True
        for i in range(n):
            # Create a list of distances between runners. Take the mininimum distance
            distances = [abs((s[i]*t) % n - (s[j]*t) % n) for j in range(n) if j != i]
            minimum_distance = min(distances)
            if minimum_distance < 1/s[i]:
                lonely = False
                break
        if lonely:
            return True
    return False

# Example input
s = [2, 3, 4, 5, 6]

# Check conjecture for this input
print(lonely_runner_conjecture(s))
```

```
True
```

Problem 10

The Cramér conjecture:

The conjecture was first proposed by the mathematician Harald Cramér in 1936, and it concerns the distribution of prime numbers.

Specifically, Cramér conjectured that the gap between consecutive primes should be at most on the order of the logarithm of the larger prime. More precisely, if p_n denotes the nth prime, then Cramér conjectured that

$$p_{n+1} - p_n = O((log\ p_n)^2)$$

where the notation "O" means that the left-hand side is at most a constant multiple of the right-hand side. [Granville (1995)]

Despite significant progress over the years, the Cramér conjecture remains unsolved.

Sage

The Cramér conjecture: Cramér conjectured that the gap between consecutive primes should be at most on the order of the logarithm of the larger prime.

```
# maximum number to consider
N = 6

# List of primes less than N
primes = prime_range(2, N)

# Differences between consecutive primes
difference = [primes[i+1] - primes[i] for i in range(len(primes)-1)]

# Take the maximum difference between consecutive primes
maximum_difference = max(difference)

# Cramér's conjecture bound for the maximum difference
cramer_bound = ((log(primes[-1]))**2).n()

print(f"The list of primes: {primes}")
print(f"Maximum difference between consecutive primes less than {N}: {maximum_difference}")
print(f"Cramér's conjecture bound: {cramer_bound}")
```

```
The list of primes: [2, 3, 5]
Maximum difference between consecutive primes less than 6: 2
Cramér's conjecture bound: 2.59029039398023
```

Bibliography

Beal, Andrew. 1997. A generalization of fermat's last theorem: the beal conjecture and prize problem. *Notices of the AMS* 44(11).

Browkin, Jerzy. 2000. The abc-conjecture. *Number theory* 75–105.

Cohn, John HE. 1967. Square fibonacci numbers. In *Exploring university mathematics*, 69–82. Elsevier.

Dris, Jose Arnaldo B. 2012. Solving the odd perfect number problem: some old and new approaches. *arXiv preprint arXiv:1204.1450* .

Dunham, William. 2013. A note on the origin of the twin prime conjecture. *Notices of the International Consortium of Chinese Mathematicians* 1(1). 63–65.

Granville, Andrew. 1995. Harald cramér and the distribution of prime numbers. *Scandinavian Actuarial Journal* 1995(1). 12–28.

Ionascu, Eugen J and Andrew Wilson. 2010. On the erdos-straus conjecture. *arXiv preprint arXiv:1001.1100* .

Machado, José A Tenreiro, Alexandra Galhano and Daniel Cao Labora. 2021. A clustering perspective of the collatz conjecture. *Mathematics* 9(4). 314.

Pandey, Ram Krishna. 2009. A note on the lonely runner conjecture. *J. Integer Seq* 12(2511224). 1233–11026.

Wang, Yuan. 2002. *The goldbach conjecture*, vol. 4. World scientific.