A generalised vision for perfect numbers and an introduction to new cryptography method

Nasr-allah Hitar

January 2023

Abstract

in this paper we will show that some idea for perefect numbers , and many other euclidean conjectures.

1 Introduction

a perfect numbers is a number which is equal to the sum of it's proper divisors,this numbers are very rare than primes but somehow those numbers are useless than primes , and also there is no generalised formula for perfect numbers and also primes.

2 Lemma's and notations

let $M_p = 2^p - 1$ when M_p is a prime we say that M_p **is a mersenne prime.**

Lemma 1. *if* $M_p \in \mathbb{P} \Longrightarrow p \in \mathbb{P}$

Lemma 2 (Euler-Euclide lemma:)**.** *if* $M_p \in \mathbb{P} \Leftrightarrow 2^{p-1}(2^p-1)$ *is perfect*

Proof. let $M_p \in \mathbb{P}$ so the proper possible divisors of $2^{p-1}(2^p - 1)$
$\mathbb{D}(2^{p-1}(2^p - 1)) = \{\ 2^k \mid 0 \leq k \leq p - 1\} \cup (2^p - 1) \cdot \{\ 2^k \mid$

$0 \leq k \leq p-2\}$
so

$$\sum_{i \in \mathbb{D}} i = \sum_{i=0}^{p-1} 2^k + (2^p-1) \sum_{k=0}^{p-2} 2^k = (2^p-1)+(2^p-1)(2^{p-1}-1) = (2^p-1)2^{p-1}$$

QED.
the other implication is trivial . $\square$

Conjecture 1 (euclide)**.** *Every perfect writen as to form of* $2^{p-1}M_p$

Conjecture 2 (as a result of conjecture2)**.** *every perfect number is a even one.*

we could use those conjecture to simplify complexity of certains computer sciences problems about perfect number ,as a result the complexity of perfect is depend on the complexity of checking primes

Proposition 1. *Odd perfect numbers are greater than* 10^{1500} this result was proved by a two person **by Pascal Ochem and Michaël Rao** from Americain mathematical society.
I will work in my future research in Cryptography using perfect numbers because I thought that can be used because they're rare than primes
here is a program of python to check perfect number

```
def checkPerfect(n):
  S = 1
  i=2
  while(i*i<=n):
    if(n%i==0):
      S+=i +(n/i)
```

```
    i+=1
  if(S==n):
    return True
  return False
```

Proof. in this proof I will use a very known theorem ,

Lemma 3. $(\forall n \in \mathbb{N})(\exists p \in \mathbb{P}) : p \leq \sqrt{n}$ $\square$

so all $p \in]|1, \sqrt{n}|] \Longrightarrow 2 \leq p \leq \sqrt{n}$
$\Longrightarrow \frac{1}{\sqrt{n}} \leq \frac{1}{p} \leq 1 \Rightarrow \sqrt{n} \leq \frac{n}{p} \leq n$
$\therefore 1 \leq p \leq \sqrt{n} \Rightarrow \sqrt{n} \leq \frac{n}{p} \leq n$
In other word all divisor will counted while running the algorithm.
or we could use euclide-euler lemma.

```
from math import floor
from math import log
from sympy import randprime , isprime
MersennePrimes = []
PerfectNumbers = []
def is_prime(n):
  i=2
  while(i*i<=n):
    if(n%i==0):
      return False
  return True
def puissance_rapid(a,b):
  if(b==0):
    return 1
  r = puissance_rapid(a,b//2)
  r*=r
  if(b%2):
    return r*a
  return r
n=1000
for i in range(2,n):
  N = puissance_rapid(2,i)-1
  if(isprime(N)):
    MersennePrimes.append(N)
for i in MersennePrimes:
  N = log(i+1,2)
  PerfectNumbers.append(puissance_rapid(2,N-1)*i)
print(PerfectNumbers)
```

Update !!!

```
    from math import log
from sympy import print_latex
def puissance_rapid(a,b):
  if(b==0):
    return 1
  r = puissance_rapid(a,b//2)
  r*=r
  if(b%2):
    return r*a
  return r
Mersenne = [2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607,
                                    1279, 2203, 2281, 3217, 4253,
                                   4423, 9689, 9941, 11213, 19937,
                                   21701, 23209, 44497, 86243,
                                   110503, 132049, 216091, 756839,
                                   859433, 1257787]#, 1398269,
                                   2976221, 3021377, 6972593,
                                   13466917, 20996011, 24036583,
                                   25964951, 30402457, 32582657,
                                   37156667, 42643801, 43112609,
                                   57885161, 74207281, 77232917,
                                   82589933]
PerfectNumbers = []
MersennePrimes = []
for i in range(0,len(Mersenne)):
  MersennePrimes.append((puissance_rapid(2,Mersenne[i])) -1)
for i in MersennePrimes:
  N = log(i+1,2)
  PerfectNumbers.append(puissance_rapid(2,N-1)*i)
print((PerfectNumbers))
```

so using this algorithm we could get more big perfect numbers greater than (Update !!!!!) $10^7 57,263$

3 References'

[1]- Euclid (1956), The Thirteen Books of The Elements, Translated with introduction and commentary by Sir Thomas L. Heath, Vol. 2 (Books III–IX) (2nd ed.), Dover, pp. 421–426. See in particular Prop. IX.36.

- Euler, Leonhard (1849), "De numeris amicibilibus" [On amicable numbers], Commentationes arithmeticae (in Latin), vol. 2, pp. 627–636. Originally read to the Berlin Academy on February 23, 1747, and published posthumously. See in particular section 8, p. 88.

- R. P. Brent, G. L. Cohen, and H. J. J. te Riele, Improved techniques for lower bounds for odd perfect numbers, Math. Comp. 57 (1991), no. 196, 857–868. MR 1094940, DOI 10.1090/S0025-5718-1991-1094940-3.

- P.Ochem , Micheal rao Odd perfect numbers are greater than 10^{1500},Math. Comp. 81 (2012), 1869-1877