Integer Factorization

Mehmet Acar

Advisor: Tülay Ayyıldız Akoğlu

5 April 2023

Project Definition

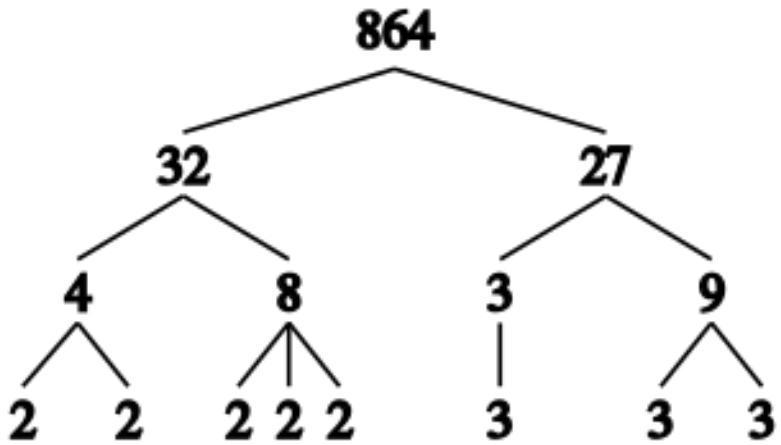
In number theory, integer factorization is the decomposition, when possible, of a positive integer into a product of smaller integers. If the factors are further restricted to be prime numbers, the process is called prime factorization, and includes the test whether the given integer is prime (in this case, one has a "product" of a single factor).

Project Definition

Factoring a positive integer n means finding positive integers p and q such that the product of p and q equals n and such that both p and q are greater than 1, p and q are called the factors of n and n = p . q is called a factorization of n.

Project Definition

To illustrate integer factorization into prime factors, take for example the integer 864 which can be factored into two numbers 32 and 27, 32 can in turn be factorized into 4 and 8, whereas 27 can be factorized into 3 and 9, thus we have: $4 = 2 \times 2 = 2^2$, $8 = 2 \times 2 \times 2 = 2^3$ and $3 = 3^1$, $9 = 3 \times 3 = 3^2$. Then if we collect the factors together the prime factorization of 864 can be written as $2^5 \times 3^3$. This can also be illustrated by the image below



Project Design - Trial Division Algorithm

Trial division is the most laborious but easiest to understand of the integer factorization algorithms. The essential idea behind trial division tests to see if an integer n, the integer to be factored, can be divided by each number in turn that is less than n. For example, for the integer n = 12, the only numbers that divide it are 1, 2, 3, 4, 6, 12. Selecting only the largest powers of primes in this list gives that $12 = 3 \times 4 = 3 \times 2^2$.
Example: $18 = 2 \times 3 \times 3$ So prime decomposition of 18 is 2, 3, 3

Trial Division Algorithm Example

```
def trial_division(n: int) :
    a = []
    f = 2
    while n > 1:
        if n % f == 0:
            a.append(f)
            n //= f
        else:
            f += 1
    return a
num = int(input("Enter number: "))
a = trial_division(num)
print("Prime factorization of number " + str(num) + ": " + str(a)  )
```

```
Enter number: 36
Prime factorization of number 36: [2, 2, 3, 3]
```

Project Design - Fermat's Factorization Method

- Fermat's factorization method relies on the fact that every odd number can be represented as a difference of squares of two numbers. That is,
- N = X^2- Y^2 = (X + Y) * (X - Y). Here 'X' is greater than 'Y' and (x + y) and (x - y) are factors of N.
- We start with finding an integer 'K' such that K * K is greater than N.
- Then we find the difference between K * K and N. Let the difference be denoted as D.
- If D is a perfect square, then we stop. Let S be the square root of D. Therefore, our answer is given by "S * S - K * K". As a result, factors of N are given by (S - K) and (S + K).

Fermat's Factorization Method Example

```
from math import ceil, sqrt

def FermatFactors(n):

        if(n<= 0):
                return [n]

        if(n % 2) == 0:
                return [n / 2, 2]

        a = ceil(sqrt(n))

        if(a * a == n):
                return [a, a]

        while(True):
                b1 = a * a - n
                b = int(sqrt(b1))
                if(b * b == b1):
                        break
                else:
                        a += 1
        return [a-b, a + b]

# Driver Code
num = int(input("Enter a number whose factors are to be found: "))
a = FermatFactors(num)
print("The factors of " + str(num) + " are " + str(a) );
```

```
Enter a number whose factors are to be found: 55
The factors of 55 are [5, 11]
```

Project Design Plan

- Until now, I searched some integer factorizations algorithms such as Trial Division, Pollard p-1 and Fermat's factorization methods.
- At the rest of of the semester, first of all I focus on selecting the algorithm and then, I am going to improve this algorithm in Python.
- Then, I prepare a graphical user interface for taking input number from user. Then, I printed output to the screen in interface.

Project Requirements - 1

- Literature review about integer factorization
- Research integer factorization algorithms
- Decide an algorithm for improvement
- Implement an improved algorithm in Python
- Preparing a GUI for user

Project Requirements - 2

Software Requirements
- I decide to the implement selected algorithm in Python which has version 3.10.1

Hardware Requirements
- No hardware requirements are needed

Success Criteria

- Factorize an integer to the prime factors with selected algorithm
- Adding some new parts to the selected algorithm logically
- GUI should be work for user

References I

[One] [Two] [Thr] [Fou]


Fermat's factorization method, https://www.codingninjas.com/codestudio/library/fermat-s-factorization-method.


Integer factorization, https://en.wikipedia.org/wiki/Integer_factorization#:~:text=In%20number%20theory%2C%20integer%20factorization,a%20product%20of%20smaller%20integers.


Integer factorization algorithms, https://iq.opengenus.org/integer-factorization-algorithms/.


Integer factorization algorithms, https://www.diva-portal.org/smash/get/diva2:1460632/FULLTEXT01.pdf.