Intro to Cryptography

Mark Anderson
Problem 7

May 16, 2019

1. Prove that for every prime not equal to 2, p mod 4 = 1 or p mod 4 = 3.
2. Congruent Primes
   - $P_{1,1} = 46062$
   - $P_{1,3} = 45981$
   - $P_{2,1} = 40547$
   - $P_{2,3} = 40638$
   - $\frac{P_{1,1}}{P_{1,3}} = 1.00176$
   - $\frac{P_{2,1}P_{2,3}}{=}.99776072$
3. Show that any prime in $P_{1,1}P_{2,1}$ can be expressed as the sum of 2 squares.

```
    #!/usr/bin/env python3
import math

def primesInRange(lower, upper):
    primes=[]
    for num in range(lower, upper + 1):
        for i in range(2, int(math.sqrt(num)) + 1):
            if num % i == 0:
                break
        else:
            primes.append(num)

    return primes

def congruency(primes, b, n):
    for number in primes:
        if (number - b) % n != 0:
            primes.remove(number)

    return primes
```

```
def egcd(a,b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, x, y = egcd(b % a, a)
        return (g, y - (b // a) * x, x)

def main():

    lowerBounds = [1000000, 10000000]
    upperBounds = [2000000, 11000000]
    #Length P11: 46062
    #Length P13: 45981
    #Length P21: 40547
    #Length P23: 40638
    P11 = congruency(primesInRange(lowerBounds[0], upperBounds[0]), 1, 4)
    print("Finished P11")
    #P13 = congruency(primesInRange(lowerBounds[0], upperBounds[0]), 3, 4)
    print("Finished P13")
    P21 = congruency(primesInRange(lowerBounds[1], upperBounds[1]), 1, 4)
    print("Finished P21")
    #P23 = congruency(primesInRange(lowerBounds[1], upperBounds[1]), 3, 4)
    print("Finished P23")

#    print("Length P11: %d\nLength P13: %d\nLength P21: %d\nLength P23:
    %d\n" % (len(P11), len(P13), len(P21), len(P23)))

    #Here is my barebones implementation for proving that each number in P11
        U P21 can be expressed as the sum of 2 squres
    #I started off attempting this problem by computing a couple of these
        numbers by hand, and they worked for each hand calculation, and I
        knew that if I had a list of all the prime numbers from [2,
        UPPERBOUND] then in that list would be 2 primes that sum up to a
        number in P11 U P21.
    #However As I started calculating these
    PU = P11 + P21
    PU = sorted(PU, reverse=True)
    #print(PU[:20])
    primes = primesInRange(2,upperBounds[1])
    print(primes)
    tally = 0
    for prime in PU[:20]:
        for a in primes:
            for b in primes:
                if a**2 + b**2 == prime:
```

```
                    tally += 1
                    print("Success")
                    print(a**2 + b**2)


    print(len(PU))
    print(tally)

if __name__ == "__main__":
    main()
```

I started out this problem by calculating a few of the numbers using brute force in python, the numbers that I tried all turned out to be true, so I assumed the theorem to be true. I started researching around and found that this is infact a theorem proven by fermat, namely "Fermats theorem of 2 squares". After reading around that theorem I was unable to deduce an efficient way to calculate the sum of these 2 squares in python, and so I immediately switched to implementing a brute force method. This method takes a list of all primes in [0, 11000000] and calculates all of the sums of these primes squared, and if it exist in the union, I increment a counter. In theory if this theorem is true, the counter should equal the length of the list containing the union, it worked for the first couple of numbers but then stalled due to having such large calculations. I was unable to find a better way to implement this other than brute force.

4. It is impossible for the sum of two squares to equal a number that is congruent to 3 mod 4 We will break this down into 4 cases to prove that this cannot be true. for all cases assume $a, b \in Z$

   - a, b are both even integers

     any even integer $a \equiv 0(mod 2) \rightarrow a^2 \equiv 0(mod 2^2)$

     Using this we can say if a,b are even, then $a^2 \equiv 0(mod 4)$ and $b^2 \equiv 0(mod 4)$, and we can show that $a^2 + b^2 \equiv (0 + 0)(mod 4)$. Thus, the only outcome for the sum of 2 even squares is $0(mod 4)$

   - a, b are both odd integers

     any odd integer $a \equiv 1(mod 2) \rightarrow a^2 \equiv 1^2(mod 2^2)$

     Using this we can say if a,b are odd, then $a^2 \equiv 1(mod 4)$ and $b^2 \equiv 1(mod 4)$ and we can show that the sum of the squares $a^2 + b^2 \equiv (1 + 1)(mod 4)$. Thus the only outcome for the sum of 2 odd squares is $2(mod 4)$

   - a is odd and b is even or b is odd and a is even

     We use the same properties as above to show that if one of the integers is odd and the other is even we get $a^2 \equiv 0(mod 4)$ and $b^2 \equiv 1(mod 4)$ so the sum of the 2 squares is $a^2 + b^2 \equiv (0 + 1)(mod 4)$

   Using the outcomes of all of these cases, we show that it covers every combination of a, b and that no combination of a,b results in $3(mod 4)$