

Winter 2021 MTH 440/540 Homework 3

Wyatt Whiting

Instructions. Due via Gradescope on Friday February 12. Solutions must be typed. Problems not marked with * are required for everyone. Problems marked with * are required for 540 students and are optional challenge problems for 440 students (they will not be graded for 440 students). Use Sage only on problems which explicitly say to use Sage; solve all other problems without Sage and show your calculations.)

17. Show that if n is a positive integer and $n \equiv 2 \pmod{4}$, then $8^n + 9^n$ is divisible by 5.

I will prove this with induction. Let's start with the base case, $n = 2$:

$$8^2 + 9^2 = 64 + 81 = 145 \equiv 0 \pmod{5}$$

Now for the induction step, we assume $8^n + 9^n \equiv 0 \pmod{5}$ where $n \equiv 2 \pmod{4}$. We then induce by replacing n by $(n + 4)$. This guarantees that $n + 4 \equiv n \pmod{4} + 4 \pmod{4} \equiv 2 \pmod{4} + 0 \pmod{4} \equiv 2 \pmod{4}$. This also doesn't "skip" any integers equivalent to 2 (mod 4), so it will cover all integers we care about.

$$\begin{aligned} 8^{n+4} + 9^{n+4} &= 8^n \cdot 8^4 + 9^n \cdot 9^4 = 8^n \cdot 4096 + 9^n \cdot 6561 \equiv 8^n \cdot 4096 \pmod{5} + 9^n \cdot 6561 \pmod{5} \\ &\equiv 8^n \cdot 1 \pmod{5} + 9^n \cdot 1 \pmod{5} \equiv 1 \cdot (8^n + 9^n) \pmod{5} \equiv 8^n + 9^n \pmod{5} \end{aligned}$$

By assumption, $8^n + 9^n \equiv 0 \pmod{5}$, so we have shown that if the condition holds for n such that $n \equiv 2 \pmod{4}$, the condition holds for $n + 4$ as well. By the principle of induction, the condition holds for all $n \in \mathbb{N}$ such that $n \equiv 2 \pmod{4}$.

18. Show that if $p \geq 5$ is prime and $a, b \in \mathbb{Z}$, then $ab^p - a^pb$ is divisible by $6p$.

We begin by observing that if a and b are both odd, then the terms ab^p and a^pb are both products of only odd numbers, so both terms will be odd. The difference of two odd terms is even, so $ab^p - a^pb$ will be even. If either a or b are even, then the terms ab^p and a^pb will both be even since having either a or b even guarantees a factor of 2. In this case, the difference $ab^p - a^pb$ is even as well. Since $ab^p - a^pb$ is even in all cases, then $2|ab^p - a^pb$.

We now want to demonstrate divisibility by 3. If either $a \equiv 0 \pmod{3}$ or $b \equiv 0 \pmod{3}$, then trivially $ab^p - a^pb \equiv 0 \pmod{3}$. If neither a nor b are divisible by 3, then we may apply Fermat's Little Theorem to say, without loss of generality, $a^{3-1} \equiv 1 \pmod{3} \implies a^2 \equiv 1 \pmod{3}$. We can raise the powers of a to see $a^3 \equiv a \pmod{3}$ and $a^4 \equiv (a^2)^2 \pmod{3} \equiv 1^2 \pmod{3} \equiv 1 \pmod{3}$, so even powers of a are 1 (mod 3). Note that since $p \geq 5$, p is guaranteed to be odd, so $p - 1$ is even and, WLOG, $a^{p-1} \equiv 1 \pmod{3}$. We now see $ab^p - a^pb = ab(a^{p-1} - b^{p-1}) \equiv ab(1 - 1) \pmod{3} \equiv ab(0) \pmod{3} \equiv 0 \pmod{3}$, so in all cases $ab^p - a^pb$ is divisible by 3.

By Fermat's Little Theorem, $b^p \equiv b \pmod{p} \implies b^p - b \equiv b - b \pmod{p} \implies b^p - b \equiv 0 \pmod{p}$, so p divides $b^p - b$. We can multiply the term by a which does not change its divisibility, so $p|ab^p - ab$. By the same line of reasoning, $p|a^pb - ab$. Since we know that p divides both $ab^p - ab$ and $a^pb - ab$, then it divides the difference $ab^p - ab - (a^pb - ab) = ab^p - a^pb$, so we now have $p|ab^p - a^pb$.

Putting these together, note that $p \neq 2, 3$, so we can say that for some x , if we have $p|x$, $2|x$, and $3|x$, then $p \cdot 2 \cdot 3 = 6p|x$. Thus, we have shown that $6p|ab^p - a^pb$.

19. Let $n \geq 1$ and let $m = 2^n - 1$.


(a) Prove that if m is prime then n is prime.

Let's consider the case when n is composite such that $n = r \cdot q$. Then the expression $2^n - 1$ can be rewritten as $2^{r \cdot q} - 1 = (2^r)^q - 1 = (2^r - 1)((2^r)^{q-1} + (2^r)^{q-2} + \cdots + 2^r + 1)$. We then see that $2^{r \cdot q} - 1$ is composite since it has $(2^r - 1)$ as a factor, which is not equal to 1. We can then say if we have $2^n - 1 = m$ where n is composite, then m is composite as well. We then consider the contrapositive of this, and we conclude that if m is prime, then n must be prime as well.

(b) Prove that if n is prime then m is either prime or a base-2 pseudoprime.

We can think about two cases of m : when m is prime, and when m is composite. If m is prime, we are done. Let's then focus on the case when m is composite and show that $2^n - 1$ is necessarily base-2 pseudoprime. If we look at the case $2^2 - 1 = 3$ is prime, we know n can only be an odd prime. By Fermat's Little Theorem, we have $2^n \equiv 2 \pmod{n}$, so $2^n - 2 \equiv 0 \pmod{n}$, so n divides $2^n - 2$.

To show that m is a base-2 pseudoprime, we must show that $2^m - 1 \equiv 1 \pmod{m}$. By the above section, we know n divides $2^n - 2$, so we can express $2^n - 2 = kn$ for some k . We can substitute this in, getting 2^{kn} . By now considering $2^{kn} - 1$, by part (a) we know that since kn is composite, $2^{kn} - 1$ has a factor of $(2^n - 1)$, so then it must be the case that $2^{kn} - 1 = 2^{2^n - 2} - 1 = 2^{2^n - 2} - 1 \equiv 0 \pmod{2^n - 2}$, so then $2^{2^n - 2} \equiv 1 \pmod{2^n - 1}$. Since $m = 2^n - 1$ is one less than a multiple of 2, then $\gcd(2, m) = 1$.

Since we have $\gcd(2, 2^n - 1) = 1$ and $2^{(2^n - 1) - 1} \equiv 1 \pmod{2^n - 1}$, then m is a base-2 pseudoprime. 

[Hint for part (a): $x^r - 1 = (x - 1)(x^{r-1} + x^{r-2} + \cdots + x + 1)$.]

20. Write a Sage program which inputs $a, b, m, n \in \mathbb{Z}$ with $m \geq 1$, $n \geq 1$, and $\gcd(m, n) = 1$, and uses Algorithm 2.2.3 to output the smallest nonnegative integer x which solves the system of congruences

$$x \equiv a \pmod{m}$$

$$x \equiv b \pmod{n}.$$

You may use Sage's `Mod(,)` command to find inverses modulo n . Note, Sage does have a built-in Chinese Remainder Theorem command, but you may not use it. Use your Sage program to find the smallest nonnegative solution $x \in \mathbb{Z}$ to the system

$$x \equiv 38755 \pmod{100001}$$

$$x \equiv 57787 \pmod{59049}.$$

```
def solve_system(a,b,m,n):
    # start with naive guess
    x = a

    # only need to check second equation
    # since first will always remain solved
    # by construction
    while(Mod(x, n) != Mod(b, n)):

        #increment by m
        x += m

    # once loops ends, system has been solved
    print(x)

# call the function with given parameters
solve_system(38755,57787,100001,59049)
```

>1238767

The smallest non-negative integer x which solves the system is 1238767.

21. In this problem you will investigate the number 1387. Do **not** attempt to factor 1387 or to directly determine whether it is prime or composite (except using your information from parts **(a)** and **(b)** below), and do not attempt to calculate $\phi(1387)$.

- (a)** Using the fast powering algorithm, compute $2^{1386} \pmod{1387}$ and $3^{1386} \pmod{1387}$. To help, you may use Sage but only for $+$, $-$, \times , \div , squaring with $\wedge 2$, and taking remainders with $\%$. Show the details of your calculations.

$$\begin{aligned} 1386 &= 1024 + 362 \\ &= 1024 + 256 + 106 \\ &= 1024 + 256 + 64 + 42 \\ &= 1024 + 256 + 64 + 32 + 10 \\ &= 1024 + 256 + 64 + 32 + 8 + 2 \end{aligned}$$

Below, all uses of \equiv are $\pmod{1387}$. First for base 2:

$$\begin{aligned} 2^2 &\equiv 4 \\ 2^4 &\equiv 4^2 \equiv 16 \\ 2^8 &\equiv 16^2 \equiv 256 \\ 2^{16} &\equiv 256^2 \equiv 65536 \equiv 347 \\ 2^{32} &\equiv 347^2 \equiv 120409 \equiv 1127 \\ 2^{64} &\equiv 1127^2 \equiv 1270129 \equiv 1024 \\ 2^{128} &\equiv 1024^2 \equiv 1048576 \equiv 4 \\ 2^{256} &\equiv 4^2 \equiv 16 \\ 2^{512} &\equiv 16^2 \equiv 256 \\ 2^{1024} &\equiv 256^2 \equiv 65536 \equiv 347 \end{aligned}$$

Now for base 3:

$$\begin{aligned} 3^2 &\equiv 9 \\ 3^4 &\equiv 9^2 \equiv 81 \\ 3^8 &\equiv 81^2 \equiv 6561 \equiv 1013 \\ 3^{16} &\equiv 1013^2 \equiv 1026169 \equiv 1176 \\ 3^{32} &\equiv 1176^2 \equiv 1382976 \equiv 137 \\ 3^{64} &\equiv 137^2 \equiv 18769 \equiv 738 \\ 3^{128} &\equiv 738^2 \equiv 544644 \equiv 940 \\ 3^{256} &\equiv 940^2 \equiv 883600 \equiv 81 \\ 3^{512} &\equiv 81^2 \equiv 1013 \\ 3^{1024} &\equiv 1013^2 \equiv 1176 \end{aligned}$$

Now we apply exponent rules:

$$\begin{aligned}
 2^{1386} &= 2^{1024+256+64+32+8+2} \\
 &= 2^{1024} \cdot 2^{256} \cdot 2^{64} \cdot 2^{32} \cdot 2^8 \cdot 2^2 \\
 &= (347)(16)(1024)(1127)(256)(4) \\
 &= 6561049083904 \\
 &\equiv 1 \pmod{1387}
 \end{aligned}$$

And for base 3:

$$\begin{aligned}
 3^{1386} &= 3^{1024+256+64+32+8+2} \\
 &= 3^{1024} \cdot 3^{256} \cdot 3^{64} \cdot 3^{32} \cdot 3^8 \cdot 3^2 \\
 &= (1176)(81)(738)(137)(1013)(9) \\
 &= 87805399740912 \\
 &\equiv 875 \pmod{1387}
 \end{aligned}$$

So $2^{1386} \equiv 1 \pmod{1387}$ and $2^{1386} \equiv 875 \pmod{1387}$.

- (b) Use Sage's `Mod(,)` command to check your answers to part (a).

```
Mod(2, 1387)^1386
```

```
> 1
```

```
Mod(3, 1387) ^ 1386
```

```
> 875
```

The hand-calculated fast powering produced the correct result in both cases.

- (c) Using your answers to part (a) and (b), answer the following true/false questions. Explain your answer in each case.

- (i) True or False: 1387 is prime.

False. Fermat's little theorem does not hold for base $a = 3$, so we can say with certainty that 1387 is not prime.

- (ii) True or False: 1387 is a base-2 pseudoprime.

True. This is because we have shown that $2^{1387-1} \equiv 1 \pmod{1387}$.

- (iii) True or False: 1387 is a base-3 pseudoprime.

False. We've shown that $3^{1387-1} \not\equiv 1 \pmod{1387}$.

- (iv) True or False: 1387 is a Carmichael number.

False. 3 is relatively prime to 1387, but $3^{1387-1} \not\equiv 1 \pmod{1387}$, so it cannot be a Carmichael number.

22. Let us declare that an integer $n \geq 2$ is a **Fermat probable prime** if the congruence $a^{n-1} \equiv 1 \pmod{n}$ holds for the first five consecutive attempts using a sequence of five randomly chosen integers a with $2 \leq a < n$. Observe that if the congruence $a^{n-1} \equiv 1 \pmod{n}$ fails to hold for even one choice of $2 \leq a < n$, then n is definitely composite (by Fermat's little theorem). Write a Sage program to run this probabilistic primality test on the following integers, declaring each to be either a **Fermat probable prime** or **composite**. Use Sage's `randrange(2,n)` command, which selects a randomly chosen integer in the specified range each time it is called. Also, be sure to use Sage's `Mod(,)` command to speed things up, with the syntax `Mod(a,n)^(n-1)` so that Sage knows to use fast exponentiation. Do not use powerful commands such as `is_prime()`.

- (a) $n = 133324441$
- (b) $n = 976303487$
- (c) $n = 6129388091$
- (d) $n = 3324344347$

```
def is_probable_prime(n):

    # run up to five times
    for i in range(5):

        # if a^(n-1) != 1 mod n -> definitely composite
        if 1 != Mod(randrange(2, n), n)^(n-1):
            return false

    # if all tests pass, it's probably prime
    return true

checknums = [133324441, 976303487, 6129388091, 3324344347]
for n in checknums:
    if is_probable_prime(n): print(n, "is probably prime")
    else: print(n, "is definitely composite")

> 133324441 is probably prime
> 976303487 is definitely composite
> 6129388091 is definitely composite
> 3324344347 is probably prime
```

23.* Write a Sage program to find all base-2 pseudoprimes less than 10000. How many are there? (You may use Sage's `is_prime()`, `Mod(,)`, and `gcd(,)` commands.)

```
a = 2 # base
count = 0 # number of base-2 pseudoprimes
# start at 4 since it's the smallest composite
for n in range(4, 10000):
    if (not is_prime(n)) and 1 == Mod(a, n)^(n - 1):
        print(n, end = " ")
        count += 1
print('\n', count)

> 341 561 645 1105 1387 1729 1905 2047 2465 2701 2821 3277 4033 4369 \
4371 4681 5461 6601 7957 8321 8481 8911
> 22
```

There are 22 base-2 pseudoprimes less than 10000.