name: kevin wong
filename: Math115 homework8
date: 04/06/2022
desc: https://courses.csail.mit.edu/6.042/spring18/mcs.pdf (Links to an external site.) please do these problems: 9.10, 9.12

9.10

Indicate **true** or **false** for the following statements about the greatest common divisor, and *provide counterexamples* for those that are **false**.

- a) If $gcd(a,b) \neq 1$ and $gcd(b,c) \neq 1$, then $gcd(a,c) \neq 1$.

  false.

  - counter example: $a = 5, b = 10, c = 2$

    $gcd(5,10) = 5 \neq 1$

    and

    $gcd(10,2) = 2 \neq 1$

    but

    $gcd(5,2) = 1$

- b) If $a|bc$ and $gcd(a,b) = 1$, then $a|c$.

  true.

- c) $gcd(a^n, b^n) = (gcd(a,b))^n$

  true.

- d) $gcd(ab, ac) = a * gcd(b,c)$

  true.

- e) $gcd(1+a, 1+b) = 1 + gcd(a,b)$

  false.

  - counter example: $a = 2, b = 1$

    $gcd(1+2, 1+1) = gcd(3,2) = 1$

    but

    $1 + gcd(2,1) = 2$

- f) If an integer linear combination of a and b equals 1, then so does some integer linear combination of a and b^2.

  true.

- g) If no integer linear combination of a and b equals 2, then neither does any integer linear combination of a^2 and b^2.

true.

9.12

Here is a game you can analyze with number theory and always beat me. We start with two distinct, positive integers written on a blackboard. Call them a and b. Now we take turns. (I'll let you decide who goes first.) On each turn, the player must write a new positive integer on the board that is the difference of two numbers that are already there. If a player cannot play, then they lose.

For example, suppose that 12 and 15 are on the board initially. Your first play must be 3, which is $15 - 12$. Then I might play 9, which is $12 - 3$. Then you might play 6, which is $15 - 9$. Then I can't play, so I lose.

- (a) Show that every number on the board at the end of the game is a multiple of $gcd(a, b)$.

  Treating this game as list data structure that starts with a and b, $b > a$. The data structure populates itself with new elements by storing the difference c, $c = ListElement1 - ListElement2$ and all list elements are less than $max(a, b)$ and greater than zero. The list also doesn't accept duplicates. This is similar to a recursively defined set or a state machine.

  The recursive step or transition to add new integers for this game has the same characteristic or preserved invariant described by the Euclid's algorithm:

  $gcd(a, b) = gcd(a, b - a),\ b > a$

  and

  $gcd(a, b) = gcd(b, rem(a, b)),\ b \neq 0$

  So, every element appended to the list is a multiple of the pair a, b and $gcd(a, b)$. ✓

- (b) Show that every positive multiple of $gcd(a, b)$ up to $max(a, b)$ is on the board at the end of the game.

  The way this game is played, assuming $b > a$, $max(a, b) = b$ already exist at the start. ✓

  Every subsequent number played has be a difference of two current numbers on the board, be distinct and must be greater than zero. So, if $b > a$, then $c = b - a$ is the first move. Then $d_1 = b - c$ or $d_2 = a - c$ is the second move and so on. This branching results in numbers that can be described in the equation below:

  $NumberOnBoard = b - q * gcd(a, b), q \in \mathbb{Z}^+$ and $b > a, b \geq NumberOnBoard \geq 0$.

  This formula represents the set of all numbers on the board that has to be played in a game. ✓

- (c) Describe a strategy that lets you win this game every time.

  Assuming b is greater than a, if $\frac{b}{gcd(a,b)}$ is odd then you should go first to win. If $\frac{b}{gcd(a,b)}$ is even then you should insist your opponent goes first so that you can win. This strategy works because we know every mulitple of the $gcd(a, b)$ less than b and greater than zero will have to be played to complete a game.

3

For each of the following pairs of numbers, do the following:

i) Find $gcd(a, b)$

ii) Express the gcd as a combination of a and b.

- extended Euclidean algorithm:

  $a = r_{-1}, b = r_0$

  $a * s_i + b * t_i = r_i$, then iterate until r_i equals to zero.

  i is the index

  q_i is the quotient

  r_i is the remainder

  s_i, t_i are the Bezout coefficients

- a) $(a, b) = (45, 36)$

i	q_i	r_i	s_i	t_i
-1	-	45	1	0
0	-	36	0	1
1	1	9	1	-1
2	4	0	-4	5

  $gcd(45, 36) = 9 = 45(1) + 36(-1)$

- b) $(a, b) = (35, 22)$

i	q_i	r_i	s_i	t_i
-1	-	35	1	0
0	-	22	0	1
1	1	13	1	-1
2	1	9	-1	2
3	1	4	2	-3
4	2	1	-5	8
5	4	0	22	-35

  $gcd(35, 22) = 1 = 35(-5) + 22(8)$

- c) $(a, b) = (331, 158)$

i	q_i	r_i	s_i	t_i
-1	-	331	1	0
0	-	158	0	1

i	q_i	r_i	s_i	t_i
1	2	15	1	-2
2	10	8	-10	21
3	1	7	11	-23
4	1	1	-21	44
5	1	0	32	-67

$gcd(331, 158) = 1 = 331(-21) + 158(44)$

4

Find the multiplicative inverse, if possible. (Most of the work for this is done in problem 3).

- a) $gcd(45, 36) = 9 \neq 1$, so there is no multiplicative inverse for $36 mod 45$.
- b) The multiplicative inverse for $22 mod 35$ is 8.
- c) The multiplicative inverse for $158 mod 331$ is 44.
- d) The multiplicative inverse for $331 mod 158$ is -21.

  or

  $-21 \equiv (158 - 21) mod(158) \equiv 137 mod(158)$

  multiplicative inverse for $331 mod 158$ can also be 137.

5

Find the smallest positive integer, x, which solves this system

$$system = \begin{cases} x \equiv_6 1 \\ x \equiv_7 5 \\ x \equiv_{19} 14 \end{cases}$$

crt:

$N = n_1 * n_2 * n_3 = 6 * 7 * 19 = 798$

$N_i = \frac{N}{n_i}$

$x = \sum_{i=1}^{3} b_i * N_i * x_i (mod N)$

b_i	N_i	$N_i = \frac{N}{N_i}$	x_i	$b_i N_i x_i$
1	6	$798/6 = 133$	$133x_1 \equiv_6 1 \rightarrow 132x_1 + 1x_1 \equiv_6 1 \rightarrow x_1 \equiv_6 1$	$1 * 133 * 1 = 133$
5	7	$798/7 = 114$	$114x_2 \equiv_7 1 \rightarrow 112x_2 + 2x_2 \equiv_7 1 \rightarrow 2x_2 \equiv_7 1 \rightarrow x_2 = 4$	$5 * 114 * 4 = 2280$

b_i	N_i	$N_i = \frac{N}{N_i}$	x_i	$b_i N_i x_i$
14	19	$798/19 = 42$	$42x_3 \equiv_1 91 \rightarrow 38x_3 + 4x_3 \equiv_1 91 \rightarrow 4x_3 \equiv_1 91 \rightarrow x_3 = 5$	$14 * 42 * 5 = 2940$

$x = 133 + 2280 + 2940(mod 798) = 5353(mod 798)$

$x = 4788 + 565(mod 798)$

$x = 565(mod 798)$

- check:
  - $565 \equiv_6 1$

    $565 mod 6 = 1 = 1 mod 6$ ✓
  - $565 \equiv_7 5$

    $565 mod 7 = 5 = 5 mod 7$ ✓
  - $565 \equiv_{19} 14$

    $565 mod 19 = 14 = 14 mod 19$ ✓

```
# calculated via python:

x = 0

while True:

    if (x % 6 == 1 % 6) and (x % 7 == 5 % 7) and (x % 19 == 14 % 19):
        print(x,"\equiv", 1 % 6, "mod 6")
        print(x,"\equiv", 5 % 7, "mod 7")
        print(x,"\equiv", 14 % 19, "mod 19")
        print("x =", x)
        break

    x += 1

---------------------------------------------------------
OUTPUT:
565 \equiv 1 mod 6
565 \equiv 5 mod 7
565 \equiv 14 mod 19
x = 565
```

END