HW5 Collaborative

CS40 Fall'21

Benjamin Cruttenden (4672440)
Bharat Kathi (5938444)
Sean Oh (4824231)
Marco Wong (4589198)

Due: Thursday, Nov 4, 2021 at 10:00PM on Gradescope

For all Collaborative HW assignments:

Collaborative homeworks may be done individually or in groups of up to 4 students. You may switch HW partners for different HW assignments. The lowest HW score will not be included in your overall HW average. Please ensure your name(s) and PID(s) are clearly visible on the first page of your homework submission.

All submitted homework for this class must be typed. Diagrams may be hand-drawn and scanned and included in the typed document. You can use a word processing editor if you like (Microsoft Word, Open Office, Notepad, Vim, Google Docs, etc.) but you might find it useful to take this opportunity to learn LaTeX. LaTeX is a markup language used widely in computer science and mathematics. The homework assignments are typed using LaTeX and you can use the source files as templates for typesetting your solutions.[1]

Integrity reminders

- Problems should be solved together, not divided up between the partners. The homework is designed to give you practice with the main concepts and techniques of the course, while getting to know and learn from your classmates.
- You may not collaborate on homework with anyone other than your group members. You may ask questions about the homework in office hours (of the instructor, TAs, and/or tutors) and on Piazza. You *cannot* use any online resources about the course content other than the text book and class material from this quarter – this is primarily to ensure that we all use consistent notation and definitions we will use this quarter.

[1] To use this template, you will need to copy both the source file (extension `.tex`) you'll be editing and the file containing all the "shortcut" commands we've defined for this class)

- Do not share written solutions or partial solutions for homework with other students in the class who are not in your group. Doing so would dilute their learning experience and detract from their success in the class.

You will submit this assignment via Gradescope (https://www.gradescope.com) in the assignment called "HW5-Collaborative".

Assigned Questions

1. (*Graded for correctness*[2]) Colors can be described as amounts of red, green, and blue mixed together[3]. Mathematically, a color can be represented as a 3-tuple (r, g, b) where r represents the red component, g the green component, b the blue component and where each of r, g, b must be a value from this collection of numbers from 0 to 255:

   $\{0, 1, 2, \ldots, 255\}$

   (a) **True** or **False**: $(1, 3, 4)$ fits the definition of a color above.

   Answer: True

   (b) **True** or **False**: $(1, 100, 200, 0)$ fits the definition of a color above.

   Answer: False

   (c) **True** or **False**: $(510, 255)$ fits the definition of a color above.

   Answer: False

   (d) **True** or **False**: There is a color (r_1, g_1, b_1) where $r_1 + g_1 + b_1$ is greater than 765.

   Answer: False

   (e) **True** or **False**: There is a color (r_2, g_2, b_2) where $r_2 + g_2 + b_2$ is equal to 1.

   Answer: True

   (f) **True** or **False**: Another way to write the collection of allowed values for red, green, and blue components is

   $$\{x \in \mathbb{N} \mid 0 \leq x \leq 255\}$$

   Answer: False

   (g) **True** or **False**: Another way to write the collection of allowed values for red, green, and blue components is

   $$\{n \in \mathbb{Z} \mid 0 \leq n \leq 255\}$$

   Answer: True

   (h) **True** or **False**: Another way to write the collection of allowed values for red, green, and blue components is

   $$\{y \in \mathbb{Z} \mid -1 < y \leq 255\}$$

   Answer: True

2. (*Graded for correctness*) For many applications in cryptography and random number generation, dividing very large integers efficiently is critical. Recall **The Division Algorithm** (zyBook 5.2): Let n be an integer and d a positive integer. There are unique integers q and r, with $0 \leq r < d$, such that $n = dq + r$. In this case, d is called the divisor, n is called the dividend, q is called the quotient, and r is called the remainder. We write $q = n$ **div** d and $r = n$ **mod** d.

   One application of the Division Algorithm is in computing the integer part of the logarithm. When we discuss algorithms in this class, we will usually write them in pseudocode or English. Sometimes we will find it useful to relate the pseudocode to runnable code in a programming language. We will typically use C++ for this.

[2]This means your solution will be evaluated on the correctness only. No justification is needed.

[3]This RGB representation is common in web applications. Many online tools are available to play around with mixing these colors, e.g. https://www.w3schools.com/colors/colors_rgb.asp

Calculating log in pseudocode

```
1  procedure log(n: a positive integer)
2  r := 0
3  while n > 1
4    r := r+1
5    n := n div 2
6  return r {r holds the result of the log operation}
```

Calculating log in C++

```
1  int log(int n) {
2    if (n < 1) {
3      cerr<<``Illegal ArgumentException''<<endl;
4    }
5    int result = 0;
6    while(n > 1) {
7      result = result + 1;
8      n = n / 2;
9    }
10   return result;
11 }
```

(a) Calculate 2027 **div** 20. *You may use a calculator if you like.*

Answer: 101

(b) Calculate 2027 **mod** 20. *You may use a calculator if you like.*

Answer: 7

(c) Calculate (-347) **div** 5 *You may use a calculator if you like.*

Answer: -69

(d) Calculate (-347) **mod** 5 *You may use a calculator if you like.*

Answer: 3

(e) How many different possible values of r (results of taking n **mod** d) are there when n is any positive integer and d is 20?

Answer: 20 different values $0 <= r < 20$

(f) What is the smallest positive integer n which can be written as $16q + 9$ for q an integer?

Answer: 9, when q = 0

3. Prove or disprove each of the following statements.

(a) For all integers a, b, and c, if $a \mid b$ and $a \mid c$, then $a \mid (b + c)$ 9, when q = 0

Answer: By direct proof. Assuming a, b, and c are all integers. Also, a divides b so $b = ka$ for some integer k and since a divides c, then $c = ja$ for some integer j. Plugging these into the expression $b + c = (ka) + (ja) = a(k + j)$. Since $(k + j)$ is an integer multiple of a, then $a|(b + c)$.

(b) For all integers a, b, and c, if $a \mid b$ and $a \mid c$, then $a \mid bc$

Answer: Towards a direct proof, assume $a|b$ and $a|c$ to show that $a|bc$, or that $bc = ka$ for some integer k. By definition,$b = ia$ for some integer i and $c = ja$ for some integer j. Plugging in b and c into bc, we get: $bc = (ia)(ja) = (ija)a = ka$. Since i,j, and a are all integers, $k = (ija)$ is also an integer. Since bc can be expressed as an integer times a, $a|bc$.

(c) For all integers a, b, and c, if $a \mid b$ or $a \mid c$, then $a \mid bc$

Answer: Let a,b,c be arbitrary integers. Towards a proof by cases. Let case 1 be $(a|b) \wedge \neg(a|c)$. Let case 2 be $\neg(a|b) \wedge (a|c)$. We want to show for all cases that $a|bc$.

Case 1:
Assume that $a|b$, to show $a|bc$. Since $a|b$, $b = ka$ for some integer k. Plugging in b to bc:
$bc = (ka)c = (kc)a$

Since k and c are integers, kc is also an integer. Since bc can be expressed as an integer times a, $a|bc$.

Case 2:
Assume that $a|c$, to show $a|bc$. Since $a|c$, $c = ja$ for some integer j. Plugging in c to bc:
$bc = b(ja) = (jb)a$
Since j and b are integers, jb is also an integer. Since bc can be expressed as an integer times a, $a|bc$.

4. For each of the following inputs: (a) Use the template of the table provided above to trace Euclid's algorithm, (b) find the gcd of the two numbers by writing the output of the algorithm, (c) express the gcd as a linear combination of the two inputs.

Euclidean algorithm

```
1  procedure gcd(x,y: positive integers)
2  a := x
3  b := y
4  if a > b
5     swap a and b
6  while a ≠ 0
7     r := b mod a
8     b := a
9     a := r
10 return b {gcd(x,y) = b}
```

Template of table to trace Euclid's algorithm

x	y	r	a	b	$a \neq 0$?

$gcd(x, y) =?$

(i) gcd(81, 65)

Answer: .

x	y	r	a	b	$a \neq 0$?
81	65	16	16	65	T
81	65	1	1	16	T
81	65	0	0	1	F

(b) 1. b was returned, and looking at the trace of the algorithm b ended as 1.
(c) 1 = -4*81 +5*65

(ii) gcd(279, 77)

Answer: .

x	y	r	a	b	$a \neq 0$?
279	77	48	48	77	T
279	77	29	29	48	T
279	77	19	19	29	T
279	77	10	10	19	T
279	77	9	9	10	T
279	77	1	1	9	T
279	77	0	0	1	T

(b) 1. b was returned, and looking at the trace of the algorithm b ended as 1.
(c) 1 = -8*279 + 29*77

5. For each x and n, find the multiplicative inverse **mod** n of x. Your answer should be an integer s in the range 0 through $n-1$. Check your solution by verifying that sx **mod** $n = 1$.

(a) $x = 35, n = 48$

Answer: .

x = 35 , n = 48
gcd(48,35)
Euclidean Sequence of gcd(48,35) yields:
48 35 13 9 4 1 0
so gcd(48,35) is 1, we can find multiplicative inverse mod 48 of 35 then by:

Linear combination: 1 = 11*35 - 8*48

11 * 35 mod 48 = 1
Multiplicative inverse mod is 11

11*35 mod 48 = 1

(b) $x = 34, n = 55$

Answer: .

x= 34, n = 55
The Euclidean algorithm for gcd(34,55) in the left column:

55 = 1*34 + 21	21 = 55 - 1*34
34 = 1* 21 + 13	13 = 34 - 1*21
21 = 1*13 +8	8 = 21 - 1*13
13 = 1*8 + 5	5 = 13 - 1*8
8 = 1*5 + 3	3 = 8 - 1*5
5 = 1*3 + 2	2 = 5 - 1*3
3 = 1*2 + 1	1 = 3 - 1*2
2 = 2*1 + 0	

gcd(34,55) = 1
Using the right column, plug in values into 1 = 3 - 1*2 to get:
Linear Combination: 1 = 34*34 - 21*55
The inverse mod n of x is 34
Check: (34)(34) mod (55) = 1
Thus, s = 34