CSE 311: Foundations of Computing I Spring 2026

Homework 3 Part 2

Due: Friday, May 1st by 6:00 PM

Instructions

Write up carefully argued solutions to the following problems. Each solution should be clear enough that it can explain (to someone who does not already understand the answer) why it works. However, you may use results from lecture, the reference sheets, and previous homework without proof.

Collaboration policy. You are required to submit your own solutions. You are allowed to discuss the homework with other students. However, the **write up** must clearly be your own, and moreover, you must be able to explain your solution at any time. We reserve ourselves the right to ask you to explain your work at any time in the course of this class.

Late policy. You have a total of **three** late days during the quarter, but you can only use one late day on any one homework, giving an additional day on both parts. Please plan ahead.

Solutions submission. Submit your solution via Gradescope. In particular:

- Each numbered task should be solved on its own page (or pages). Do not write your name on the individual pages. (Gradescope will handle that.)
- When you upload your pages, make sure each one is **properly rotated**. If not, you can use the Gradescope controls to turn them to the proper orientation.
- Follow the Gradescope prompt to **link tasks to pages**.
- You are not required to typeset your solution, but your submission must be **legible**. It is your responsibility to make sure solutions are readable — we will *not* grade unreadable write-ups.

Task 1 – Euclidean, My Dear Watson [10 pts]

We say that an equation is in "**standard form**" if it looks like $Ax \equiv_n B$ for some constants A, B, and n. The first equation below is in standard form, but the second is *not*.

Solve each of the below modular equations by following these steps, showing your work as described next.

1. If the modular equation is *not* in standard form, then **transform** it into standard form.

   Show the sequence of operations, either adding to both sides or simplifying (e.g., algebraically modifying terms on individual sides as done in lecture).

2. **Calculate** *one solution* to the modular equation in standard form using the Extended Euclidean Algorithm.

   Show your work by writing out the sequence of quotients and remainders, the resulting tableau, and the sequence of substitutions needed to calculate the relevant multiplicative inverse. Then, show how multiplying the initial equation on both sides by the multiplicative inverse gives you a solution to the equation.

3. **State** *all integer solutions* to the modular equation in standard form.

   Your answer should be of the form "$x = C + Dk$ for any integer k", where C and D are integers with $0 \leqslant C < D$.

4. If the original modular equation was *not* in standard form, justify briefly (in one sentence) why the solutions to the equation in standard form are the same as the solutions to the original equation.

5. Show that there is some solution $z \in \mathbb{Z}$ such that $z \geqslant 1000$.

a) $7x \equiv_{38} 5$

b) $62x - 6 \equiv_{50} 4 - 25x$

Task 2 – Sum Kind of Wonderful [10 pts]

a) Prove, by induction, that

$$\sum_{i=0}^{n} \left(5 \cdot 6^i + 3\right) = 6^{n+1} + 3n + 2$$

holds for all integers $n \geqslant 0$.

Write an **English** proof, following the template given in lecture.

b) Prove, by induction, that

$$3^n \geqslant 2n + 1$$

holds for all integers $n \geqslant 1$.

Write an **English** proof, following the template given in lecture.

Task 3 – Barking Up the Strong Tree [10 pts]

When you first learned **recursion** in CSE 123, a mysterious person gave you the following recursive Java method and claimed that it behaves like the natural-number version of Math.pow():

```
int mysteriousPow(int b, int m) {  /* Assumes: b >= 1 and m >= 0 */
    if (m == 0) {
        return 1;
    } else if (m == 1) {
        return b;
    } else {
        return (b - 1) * mysteriousPow(b, m - 1) + b * mysteriousPow(b, m - 2);
    }
}
```

You wrote some tests and realized that this method might be correct, but you didn't know how to prove it rigorously...until you are taking CSE 311 and learn strong induction! Now, let's try to prove the correctness of this method. Not sure what I mean? Let's put it another way:

Let b be a positive integer. The function $f(m)$ is defined for all integers $m \geqslant 0$ recursively as follows:

$$\begin{aligned} f(0) &= 1 \\ f(1) &= b \\ f(m) &= (b-1) \cdot f(m-1) + b \cdot f(m-2) \qquad \text{if } m \geqslant 2 \end{aligned}$$

Use strong induction to prove that the following holds for all integers $n \geqslant 0$:

$$f(n) = b^n$$

Write an **English** proof, following the template given in lecture.

Task 4 – Optional Practice Problems (Extra Credit)

The following problems are optional and do not need to be submitted. However, you may submit solutions and receive a small amount of extra credit for any 5 (of the 12) subparts, graded solely on completion. For the maximum extra credit score, at least 5 subparts should be submitted, including at least one subpart from each of the three sections.

a) Find solutions to each of the following modular equations.

i) $15x \equiv_{28} 14$

ii) $13x - 3 \equiv_7 x$

b) Write an English proof for the following claims. It should not be necessary to use induction.

i) Let m and n be positive integers, with $\gcd(m, n) = 1$. Prove the following claim: for any integers a and b, there exists an integer x such that $x \equiv_m a$ and $x \equiv_n b$.[1] **Hint**: Apply Bézout's theorem.

ii) Let n and c be positive integers. For any integers a and b, if $a \equiv_n b$, then $ca \equiv_{cn} cb$. (Note that the subscript has changed from n to cn.)

iii) Let n and k be positive integers, with $\gcd(n, k) = 1$. If $ka \equiv_n kb$, then $a \equiv_n b$, for any integers a and b. **Hint**: You can use the facts proven on slide 82 of topic 3.

iv) For any positive integer a, $\gcd(a, a + 1) = 1$.

v) For any positive integers a and b, $\gcd(a, b) = \gcd(b, a)$

c) Prove the following claims using induction. Use the English proof template given in lecture. (You must decide whether to use weak or strong induction.)

i) $6 \mid (7^n - 1)$ for all integers $n \geqslant 1$.

ii) Every positive integer is either even or odd.

iii) Every amount of postage of $\geqslant 12$ cents can be formed using only 4-cent and 5-cent stamps. That is, for every integer $n \geqslant 12$, there exist non-negative integers a and b such that $4a+5b = n$.

iv) $2^n > n^2$ for all integers $n \geqslant 5$.

v) Consider a function g defined for all integers $n \geqslant 1$ by $g(1) = 1$, $g(2) = 3$, and $g(n) = 3g(n - 1) - 2g(n - 2)$ for $n \geqslant 3$. Prove that $g(n) = 2^n - 1$ for all $n \geqslant 1$.

[1] This claim is a simple version of (the existence part of) the Chinese Remainder Theorem.