CSE 311: Foundations of Computing I Winter 2025

Problem Set 5

Due: Wednesday, February 12th by 11:00pm

Instructions

Solutions submission. You must submit your solution via Gradescope. In particular:

- Submit a *single* PDF file containing your solutions to Task 1 and 3–6 (and optionally 7). Follow the prompt on Gradescope to link tasks to your pages.
- The instructions for submitting Task 2 appear below that individual problem.

Task 1 – A Wink and a Mod [26 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 latter two are *not*.

Solve each of the 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, then **transform** the modular equation in standard form back into the original. As done in Step 1, show the sequence of operations. [1]

a) $17x \equiv_{39} 4$

b) $23x - 7 \equiv_{61} 5x - 2$

c) $4(3x + 2) \equiv_{47} 5 - 4x$

[1] Steps 1 and 4 combined prove that the original equation and the one in standard form have identical solutions.

Task 2 – Winnie the Two [10 pts]

We can use mathematical induction to prove that $P(n)$ holds for integers $n \geqslant b$ via the following rule:

Induction

$$\frac{P(b) \quad \forall n\,(P(n) \to P(n+1))}{\therefore \quad \forall n\,((n \geqslant b) \to P(n))}$$

In other words, if we know that $P(b)$ holds and we know that, whenever $P(n)$ holds, so does $P(n+1)$, then it must be the case that $P(n)$ is true for all integers $n \geqslant b$.

To gain some familiarity with this rule (called "`induction`" in Cozy), let's do a proof...

Prove, by induction, that $2 \mid n(n+1)$ holds for all integers $n \geqslant 0$.

Write a **formal** proof that the claim holds.

> Submit and check your formal proof here:
>
> http://cozy.cs.washington.edu
>
> You **must also** include your solution (as a screenshot, typeset LaTeX, or rewritten by hand) in the PDF you submit to **Gradescope**.

It will require a little more Cozy knowledge to complete this problem...

Suppose that we were asked to prove $\exists k\,(2a = a + k)$. Cozy will let us write down $2a = a + a$ using the `algebra` rule. If we then try to use `intro exists` to replace the last a, it will replace all "a"s giving us $\exists k\,(2k = k + k)$, which is not what we want:

```
1. 2*a = a + a                algebra (2*a = a + a)
2. exists k, 2*k = k + k      intro exists 1 {a} k
```

By default, Cozy replaces *all* instances of the expression you give it (e.g., `a`) with the new existential variable (e.g., `k`). However, it is also possible to only replace *some* instances.

To do so, you pass an optional fourth argument to `intro exists` that gives it the exact result that you want it to produce.[2] Here is how we would do it in the previous example:

```
1. 2*a = a + a              algebra (2*a = a + a)
2. exists k, 2*a = a + k    intro exists 1 {a} k (exists k, 2*a = a + k)
```

This allows us to produce the statement, $\exists k\,(2a = a + k)$, with only the last "a" replaced by "k", which is exactly the one we wanted.

[2] Cozy will double check that the proposition you give it is the result of replacing *some* instances of that expression by the existential variable. It won't just let you write down any proposition you want!

Task 3 – Sum Kind of Wonderful [20 pts]

Prove, by induction, that

$$\sum_{i=0}^{n}(10(11)^i + 1) = (11)^{n+1} + n$$

holds for all integers $n \geqslant 0$.

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

Task 4 – Less To Impress [20 pts]

Prove, by induction, that $n2^n < 2^{2n}$ holds for all integers $n \geqslant 2$.

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

Task 5 – In the Strong Place, At the Strong Time [20 pts]

The function $f(m)$ is defined for all integers $m \geqslant 0$ recursively as follows:

$$\begin{aligned} f(0) &= 1 \\ f(1) &= 3 \\ f(m) &= 2 \cdot f(m-1) + 3 \cdot f(m-2) \qquad \text{if } m \geqslant 2 \end{aligned}$$

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

$$f(n) = 3^n$$

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

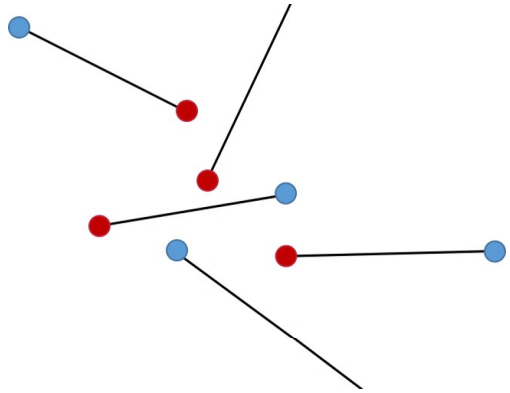
Task 6 – Modding Off [8 pts]

a) Compute $3^{345} \bmod 100$ using the efficient modular exponentiation algorithm. Show all intermediate results.

b) How many multiplications does the algorithm use for this computation? (Assume that we do not need to perform a multiplication to calculate $3^1 = 3$ since we know that $x^1 = x$ for any x.)

Task 7 – Extra Credit: Match Me If You Can [0 pts]

In this problem, you will show that given n red points and n blue points in the plane such that no three points lie on a common line, it is possible to draw line segments between red-blue pairs so that all the pairs are matched and none of the line segments intersect. Assume that there are n red and n blue points fixed in the plane.



A *matching* M is a collection of n line segments connecting distinct red-blue pairs. The *total length* of a matching M is the sum of the lengths of the line segments in M. Say that a matching M is *minimal* if there is no matching with a smaller total length.

Let IsMinimal(M) be the predicate that is true precisely when M is a minimal matching. Let HasCrossing(M) be the predicate that is true precisely when there are two line segments in M that cross each other.

Give an argument in English explaining why there must be at least one matching M so that IsMinimal(M) is true, i.e.

$$\exists M \text{IsMinimal}(M))$$

Give an argument in English explaining why

$$\forall M(\text{HasCrossing}(M) \rightarrow \neg \text{IsMinimal}(M))$$

Then, use the two results above to give a proof of the statement:

$$\exists M \neg \text{HasCrossing}(M).$$