CMSC 27200 - Spring 2024 — Lorenzo Orecchia
due 3/29/24 via Gradescope

H2.

- The assignment is due at Gradescope on 3/29/24.
- A LaTeX template will be provided for each homework. You are strongly encouraged to type your homework into this template using LaTeX. If you are writing by hand, please fill in the solutions in this template, inserting additional sheets as necessary. This will help facilitate the grading.
- You are permitted to discuss the problems with up to 2 other students in the class (per problem); however, *you must write up your own solutions, in your own words*. Do not submit anything you cannot explain. If you do collaborate with any of the other students on any problem, please list all your collaborators in the appropriate spaces.
- Similarly, please list any other source you have used for each problem, including other textbooks or websites.
- *Show your work.* Answers without justification will be given little credit.
- Your homework is *resubmittable*. Please refer to the course syllabus on Canvas for a more detailed description of this. For any problem that you have not changed from your last submission, please make sure to indicate this in your submission to help our graders grade faster.

PROBLEM 1 *Consider the recursive version of the gcd algorithm given in class and in the DPV textbook.*

1. *How much space, i.e., memory allocation, does this algorithm require as stated?*
2. *Write pseudocode for an iterative version of the algorithm that reduces the space requirement to $O(n)$ bits.*

Collaborators: Zack Hassman

Solution:

1. $O(n^2)$

   There are at most n recursive calls

   Each call stores an n-bit integer a, and an m-bit integer b, where $m \leq n$.

2. Iterative algorithm to compute $gcd(a, b)$ in linear time:

$gcd(a, b)$
Input: $a, b \in \mathbb{N}, a \geq b \geq 0$
Output: $d \in \mathbb{N}$
1: **while** $b \neq 0$ **do**
2: $t \leftarrow a$
3: $a \leftarrow b$
4: $b \leftarrow t \mod b$
5: **end while**
6: **return** a

PROBLEM 2 *Let F_i be the ith Fibonacci number. Consider the execution of the recursive* gcd *algorithm on inputs (a, b) with $a > b \geq 0$. Show that if $b \leq F_k$ for some k, then the algorithm makes at most k recursive calls.*

Collaborators: Zack Hassman

Solution:
Proof.
We use strong induction on k.
Base Case
If $k = 0$ then $b \leq 0$ so the algorithm returns a after 0 recursive calls.
Inductive Case
Suppose the algorithm takes at most $k-1$ recursive calls for all $b \leq F_{k-1}$.
Consider $b \leq F_k$. The algorithm makes a recursive call to $gcd(b, a \mod b)$, which makes its own recursive call to $gcd(a \mod b, b \mod (a \mod b))$.
Now, $[a \mod b] + [b \mod (a \mod b)] \leq b$. Since $b \leq F_k = F_{k-1} + F_{k-2}$, either $a \mod b \leq F_{k-1}$ or $b \mod (a \mod b) \leq F_{k-2}$
If $a \mod b \leq F_{k-1}$, then $gcd(b, a \mod b)$ takes at most $k-1$ calls by the inductive hypothesis. So $gcd(a, b)$ takes $k - 1 + 1 = k$ calls.
If $b \mod (a \mod b) \leq F_{k-2}$ then $gcd(a \mod b, b \mod (a \mod b))$ takes $k-2$ calls by the inductive hypothesis. So $gcd(a, b)$ takes $k - 2 + 2 = k$ calls.
□

Problem 3 (Programming: Probabilistic Testing and Primality) *Follow this Google Colab template link to implement Fermat primality testing.* ***Do not attempt to edit the linked file. Create your own copy instead by clicking File → Save a copy in Drive****. When you have completed it, download it as a Jupyter Notebook and submit it to the appropriate Gradescope assignment.*

Collaborators: Zack Hassman

Solution: See Gradescope

PROBLEM 4 *Wilson's theorem says that a number N is prime if and only if*

$$(N-1)! \equiv -1 (\text{mod } N).$$

In this problem, we will walk through writing a formal proof for this theorem.

(a) *We can prove Wilson's Theorem by proving three lemmas:*

Lemma 1 *If p is prime, only 1 and $p-1$ are their own inverses modulo p.*

Lemma 2 *If p is prime, $(p-1)! \equiv -1 (\text{mod } p)$.*

Lemma 3 *If $(p-1)! \equiv_p -1$ then p is prime.*

(b) *Provide a proof of Lemma 1. You may use the fact that if p is prime, then we know every number $1 \leq x < p$ is invertible modulo p.*

(c) *Provide a proof of Lemma 2 by pairing up multiplicative inverses. You may use the fact that every number $1 \leq x < p$ has a **unique** inverse modulo p.*

(d) *Provide a proof of Lemma 3. (Hint: Note that this can only happen if:*

$$\gcd(N, (N-1)!) = 1.$$

)

(e) *We've just shown that Wilson's theorem is an if-and-only-if condition for primality. Why can't we base a primality test algorithm on Wilson's Theorem?*

Collaborators: Zack Hassman

Solution:

(b) *Proof.*

p is prime, so all $1 \leq x < p$ is invertible modulo p. If an x is is own p-modular inverse, then we have:

$$x^2 \equiv_p 1$$
$$p \mid x^2 - 1$$
$$p \mid (x-1)(x+1)$$

Since p is prime, either $p \mid x \pm 1$ so $x \pm 1 \equiv_p 0$.

These correspond to $x = 1$ and $x = p - 1$.

□

(c) *Proof.*

We claim every number in $\mathbb{Z}_p$ has a unique inverse modulo p in $\mathbb{Z}_p$.

Suppose $x \in \mathbb{Z}_p$ has an inverse $x^{-1} \notin \mathbb{Z}_p$. Then:

$$1 \equiv_p x x^{-1}$$
$$1 \equiv_p (x(x^{-1} \mod p))$$

So $x^{-1} \mod p \in \mathbb{Z}_p$ is also an inverse.

Modular inverses are involutive since $x = (x^{-1})^{-1} \pmod p$. Thus, we can partition $\mathbb{Z}_p$ into pairs of inverses, except for 1 and $p-1$, which are paired with themselves.

Using this fact:

$$
\begin{aligned}
(p-1)! &= 1 \cdot \Big(\prod_{(x,x^{-1}) \in \mathbb{Z}_p} x \cdot x^{-1} \Big) \cdot (p-1) \\
(p-1)! &\equiv_p \Big(\prod_{(x,x^{-1}) \in \mathbb{Z}_p} (x \cdot x^{-1}) \mod p \Big) \cdot ((p-1) \mod p) \\
(p-1)! &\equiv_p \Big(\prod_{(x,x^{-1}) \in \mathbb{Z}_p} 1 \Big) \cdot (-1) \\
(p-1)! &\equiv_p \Big(\prod_{(x,x^{-1}) \in \mathbb{Z}_p} 1 \Big) \cdot (-1) \\
(p-1)! &\equiv_p -1
\end{aligned}
$$

□

(d) *Proof.* We claim that $p-1$ is a p-modular inverse of $(p-1)!$. Check that this holds:

$$
\begin{aligned}
(p-1)! \cdot (p-1) &\equiv_p ((p-1)! \mod p) \cdot ((p-1) \mod p) \\
&\equiv_p (-1) \cdot (-1) \\
&\equiv_p 1
\end{aligned}
$$

Since $(p-1)!$ is invertible modulo p, $gcd(p, (p-1)!) = 1$. This implies p is indivisible by $1, \ldots, p-1$. Thus p is prime.

□

(e) The $(p-1)!$ is intractable.