Math 132: Discrete Mathematics
Problem session 4

1. The following is an inductive argument that all birds are the same color (assuming every bird has a unique color, and there are only finitely many birds). The conclusion is clearly false. Should we reject induction as a valid method of reasoning?

   Let $\mathcal{P}(n)$ be the statement "For any collection of n birds, all the birds in that collection are of the same color." $\mathcal{P}(1)$ is true, because any bird is the same color as itself. ($\mathcal{P}(0)$ is also vacuously true.) Suppose that $\mathcal{P}(n)$ is true, so *any* collection of n birds has them all of the same color. For any collection of $n+1$ birds we can find two subsets of n birds:

   $$\overbrace{*+\underbrace{*+\ldots+*}_{n \text{ birds}}}^{n \text{ birds}}+*=(n+1) \text{ birds}$$

   By the inductive hypothesis, these subsets have the same color, and since they overlap, we conclude that the entire set consists of birds of the same color, so $\mathcal{P}(n+1)$ is true.

2. Show by induction that $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$ and $\sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2$.

3. Use the fact that $\sum_{i=1}^{n+1} i^5 = \sum_{i=1}^{n+1} [(i-1)+1]^5$ to find a formula for $\sum_{i=1}^{n} i^4$. Can you see how to generalize this for $\sum_{i=1}^{n} i^5$, etc.?

4. For $a, b \in \mathbb{N}$, not both 0, let (a, b) denote the greatest common divisor. Show that $(a, b) = (b, a \mod b)$.

5. Compute $(257244, 50320)$ and $(325440, 49946)$ by hand.

6. Let p be a prime and $1 < i < p$ a natural number. Show that p divides $\binom{p}{i}$.

7. Prove Fermat's Little Theorem: If p is a prime and $a \in \mathbb{N}$ with $(p, a) = 1$, then $a^p \equiv a \mod p$.

   Try to do this both by using Problem 6 and by giving a separate, combinatorial argument.

8. Recall that Euler's totient function $\phi : \mathbb{N}_{>0} \to \mathbb{N}$ is defined by $\phi(n)$ is the number of natural numbers $i < n$ such that $(n, i) = 1$.

   (a) Show that if p is prime then $\phi(p) = p - 1$.
   (b) Compute $\phi(p^n)$ for p a prime and $n \in \mathbb{N}$.
   (c) Show that if $(n, m) = 1$ then $\phi(nm) = \phi(n)\phi(m)$.
   (d) Show that $\phi(n) = \sum_{d|n} \phi(d)$.
   (e) Show that if $(n, a) = 1$ then $a^{\phi(n)} \equiv 1 \mod n$.

9. Much of the above is conceptually simpler if we work in "modular" number systems: If $n \in \mathbb{N}_{>0}$, let $\mathbb{Z}/n$ denote the set $\{0, 1, 2, \ldots, n-1\}$, endowed with addition $\oplus$ and multiplication $\odot$, each taken mod n. Thus for $n = 6$, we have $2 \oplus 5 = 1$ as $2 + 5 = 1 \equiv 1 \mod 6$ and $2 \odot 5 = 4$ as $2 \cdot 5 = 10 \equiv 4 \mod 6$.

   (a) For $a \in \mathbb{Z}$, let $\overline{a} = a \mod n \in \mathbb{Z}/n$. Show that $\overline{a} \oplus \overline{b} = \overline{a + b}$ and $\overline{a} \odot \overline{b} = \overline{a \cdot b}$ in $\mathbb{Z}/n$.
   (b) A *unit* $a \in \mathbb{Z}/n$ is an element such that there some $b \in \mathbb{Z}/n$ with $a \odot b = 1$. Show that a is a unit in $\mathbb{Z}/n$ if and only if $(a, n) = 1$.
   (c) Can you see how to rephrase the above problems in terms of modular arithmatic?

Appliction: RSA encryption

RSA encryption (for Rivest, Shamir, and Adleman, its inventors) is an example of a *public-key* encryption scheme, i.e., a system for sending coded messages such that anyone is able to encode a message using publicly available information, but only the intended recipient can decode the message in a reasonable amount of time. The basic idea works as follows:

Suppose you want people to send you a secret message, but you don't want to meet with each sender ahead of time to work out a separate encryption scheme. Instead, you secretly pick two big prime numbers, p and q. In order for this to work, p and q should be very big—a few hundred digits should work. You do *not* tell the world what these prime numbers are; the security of the scheme depends on your being the only person who knows their identity. However, you do compute their product

$$n := p \cdot q,$$

which will be released to the public. Before doing that, compute the Euler totient function of n:

$$\phi(n) = \phi(pq) = (p-1)(q-1) = pq - p - q + 1 = n - (p + q - 1).$$

Do not release $\phi(n)$, as knowledge of this number allows decryption of all coded messages. Instead, pick some number e such that $1 \leq e < \phi(n)$ and $GCD(e, \phi(n)) = 1$. e can be significantly smaller than $\phi(n)$, but making e too small (i.e., $e = 3$) has the potential for some security risks.

You now release the pair (n, e) to the public. If someone wishes to send you a secret message (which we assume to be encoded as an integer) m, they simply compute $c := (m^e \mod n)$ and pass this along to you.

Now that you've received an encoded message, you need to figure out how to decode it. In other words, you want to be able to recover m, when all you are given is c. Luckily, you have more information than the world at large: You know $\phi(n)$, and you know that e was chosen to be relatively prime to $\phi(n)$. If you've done the above exercises, you know that there is some $d \in \mathbb{Z}/\phi(n)$ (note we're changing the modulus) such that $e \odot d = 1$ in $\mathbb{Z}/\phi(n)$, and it's possible to compute d with knowledge of e and $\phi(n)$ quickly. Now you simply compute $c^d \mod n$. Can you see why this recovers m?

Some comments:

- This entire scheme relies on the observation that $(m^e)^d = m \mod n$. In order to properly recover m from the cipher $c = m^e \mod n$, we first need that $m < n$ (since the decryption process necessarily returns an integer less than n). This should not be a problem if you chose your original primes to be big enough, or if your correspondents limit themselves to messages of reasonable length.

  Slightly more troublingly, you also should require that $GCD(m, n) = 1$. Can you see why this is important? Luckily, it's easy to check that this condition is satisfied, and any randomly chosen message n is extremely unlikely to have a common factor with n. Unluckily, if $GCD(m, n) \neq 1$, you can quickly compute the prime factorization of n, and the security of the scheme will fall apart.

- The security of the scheme relies on the fact that it is extremely hard to factor large numbers, especially if the number in question is the product of exactly two large primes.

  The practicality of the system relies on the fact that modular exponentiation, even with a big modulus n and exponents e and d, *can* be computed quickly. Even more importantly, *testing* for primality of the original p and q can be done quickly (in polynomial time), so with access to even a moderately powerful computer you should be able to generate a highly secure public encryption scheme.