CS182 Spring 2016: Homework 7

Due date: Friday, April 15, 2016 (before class).

1. (12 points) Number Theory and Cryptography

   (a) Let n be a positive integer such that $n = d_k d_{k-1} \dots d_1$ where $d_i \in \{0, 1, \dots, 9\}$ (each d_i is a digit of n). Prove that $n \mod 3 = 0$ if and only if $\left(\sum_{i=1}^{k} d_i\right) \mod 3 = 0$.

   Proof. Note that we can write n as the sum of it's digits: $n = \sum_{i=1}^{k} 10^{i-1} d_i$. Thus, we have that:

$$
\begin{aligned}
n \bmod 3 = 0 &\iff \left(\sum_{i=1}^{k} 10^{i-1} d_i\right) \bmod 3 = 0 \\
&\iff \left(\sum_{i=1}^{k} 10^{i-1} d_i \bmod 3\right) \bmod 3 = 0 \\
&\iff \left(\sum_{i=1}^{k} (10^{i-1} \bmod 3)(d_i \bmod 3)\right) \bmod 3 = 0 \\
&\iff \left(\sum_{i=1}^{k} d_i \bmod 3\right) \bmod 3 = 0 \qquad [1] \\
&\iff \left(\sum_{i=1}^{k} d_i\right) \bmod 3 = 0
\end{aligned}
$$

   Where item [1] is due to the fact that for any nonnegative integer k, $10^k \bmod 3 = 1$. □

   (b) Given an 11-digit number $x = x_{11} x_{10} \dots x_1$, this number is a USPS valid money order number if and only if $x_{11} = x_1 + x_2 + \dots + x_{10} \mod 9$. In the following questions, let Q denote a digit that has been lost. Recover the smudged digit if possible. (Note: any answer without justification/work shown will receive minimal credit)

      i. $Q1223139784$
      ii. $6702120Q988$
      iii. $213279032Q1$

   Solutions:

i. Q above represents x_{11}, so it is easy to recover Q:

$$\begin{aligned} Q &= (1+2+2+3+1+3+9+7+8+4) \bmod 9 \\ &= 38 \bmod 9 \\ &= 4 \end{aligned}$$

ii. Q here represents x_4, so we know that:

$$\begin{aligned} 6 &= (7+0+2+1+2+0+Q+9+8+8) \bmod 9 \\ &= (37+Q) \bmod 9 \\ &= (1+Q) \bmod 9 \end{aligned}$$

So $Q = 5$.

iii. Q here represents x_2, so we know that:

$$\begin{aligned} 2 &= (1+3+2+7+9+0+3+2+Q+1) \bmod 9 \\ &= (28+Q) \bmod 9 \\ &= (1+Q) \bmod 9 \end{aligned}$$

So $Q = 1$

(c) Find all solutions, if any, to the system of congruences:

$$\begin{aligned} x &\equiv 5 \pmod 6 \\ x &\equiv 3 \pmod{10} \\ x &\equiv 8 \pmod{15} \end{aligned}$$

Solution: Since $x \equiv 5 \pmod 6$, we know there exists an integer k such that $x = 5 + 6k$. Then, we have that:

$$x \equiv 5 + 6k \equiv 3 \pmod{10}$$

So we have that

$$6k \equiv 8 \pmod{10}$$

Note however that 6 does not have an inverse modulo 10, so if there is a solution, it will not necessarily be unique. Similarly, we have that:

$$x \equiv 5 + 6k \equiv 8 \pmod{15}$$

So

$$6k \equiv 3 \pmod{15} \tag{1}$$

Again, 6 does not have an inverse modulo 15, so if a solution exists, it will not necessarily be unique. Now, we have two equations:

$$6k = 8 + 10t$$
$$6k = 3 + 15s$$

This can be simplified to

$$3k = 4 + 5t$$
$$2k = 1 + 5s$$

So by subtracting the second equation from the first, we have that

$$k = 3 + 5(t - s)$$

for any $s, t \in \mathbb{Z}$. Since s, t can be any integer, $(t-s)$ can also be any integer, say w. So we have that $k = 3+5w$, and finally that $x = 5+6k = 5+18+30w = 23+30w$. To confirm that x can be any integer of that form, note that:

$$\begin{aligned} 5 &\equiv x \pmod 6 \equiv (23 + 30w) \pmod 6 \\ &\equiv [23 \pmod 6 + 30w \pmod 6] \pmod 6 \\ &\equiv 5 \pmod 6 \end{aligned}$$

$$\begin{aligned} 3 &\equiv x \pmod{10} \equiv (23 + 30w) \pmod{10} \\ &\equiv [23 \pmod{10} + 30 \pmod{10}] \pmod{10} \\ &\equiv 3 \pmod{10} \end{aligned}$$

$$\begin{aligned} 8 &\equiv x \pmod{15} \equiv (23 + 30w) \pmod{15} \\ &\equiv [23 \pmod{15} + 30 \pmod{15}] \pmod{15} \\ &\equiv 8 \pmod{15} \end{aligned}$$

2. (8 points) Basic Induction Proofs

   (a) Let $a_n = \sum_{i=0}^{n} 2(-7)^i$. Prove that for all $n \geq 0, n \in \mathbb{Z}$, $a_n = (1 - (-7)^{n+1})/4$ using induction.

   Proof. **Base Case:** $n = 0$**.** Then, $a_0 = \sum_{i=0}^{0} 2(-7)^i = 2$. Similarly, $(1-(-7)^1)/4 = 8/4 = 2$. So the base case holds.

Inductive Hypothesis: Assume that it is true for some $k \geq 0$ that $a_k = (1 - (-7)^{k+1})/4$. We show it's true for a_{k+1}. Note that $a_{k+1} = \sum_{i=0}^{k+1} 2(-7)^i$. Then,

$$\begin{aligned}
a_{k+1} &= \sum_{i=0}^{k+1} 2(-7)^i \\
&= 2(-7)^{k+1} + \sum_{i=0}^{k} 2(-7)^i \\
&= 2(-7)^{k+1} + a_k && \text{[by definition of } a_k\text{]} \\
&= 2(-7)^{k+1} + (1 - (-7)^{k+1})/4 && \text{[by Inductive Hypothesis]} \\
&= 8(-7)^{k+1}/4 + 1/4 - (-7)^{k+1}/4 \\
&= 1/4 + 7(-7)^{k+1}/4 \\
&= (1 - (-7)(-7)^{k+1})/4 && = (1 - (-7)^{k+2})/4
\end{aligned}$$

Thus, we have shown that if a_k is true, then a_{k+1} is also true. By the Principle of Mathematical Induction, the result holds for all $n \geq 0$, $n \in \mathbb{Z}$. □

(b) Suppose $a, b \in \mathbb{R}$ such that $0 < b < a$. Prove using induction that for any $n \in \mathbb{Z}^+$, we have $a^n - b^n \leq na^{n-1}(a - b)$.

Proof. **Base Case:** $n = 1$**.** Then, $a^1 - b^1 = a - b = (a - b) * 1 * a^0$.

Inductive Hypothesis: Assume that for $k \geq 1$, the claim holds true. That is, $a^k - b^k \leq ka^{k-1}(a - b)$. Note that

$$a^k - b^k = (a - b)(a^{k-1} + a^{k-2}b + \cdots + ab^{k-2} + b^{k-1})$$

So we have that

$$\begin{aligned}
& a^k - b^k \leq ka^{k-1}(a - b) \\
\iff & (a - b)(a^{k-1} + a^{k-2}b + \cdots + ab^{k-2} + b^{k-1}) \leq ka^{k-1}(a - b) \\
\iff & (a^{k-1} + a^{k-2}b + \cdots + ab^{k-2} + b^{k-1}) \leq ka^{k-1}
\end{aligned}$$

Then, consider $a^{k+1} - b^{k+1}$:

$$\begin{aligned}
a^{k+1} + b^{k+1} &= (a - b)(a^k + a^{k-1}b + \cdots + ab^{k-1} + b^k) \\
&= (a - b)[a(a^{k-1} + a^{k-2}b + \cdots + ab^{k-2} + b^{k-1}) + b^k] \\
&\leq (a - b)[a(ka^{k-1}) + b^k] && \text{[Inductive Hypothesis]} \\
&= (a - b)(ka^k + b^k) \\
&\leq (a - b)(ka^k + a^k) && [b < a \to b^k < a^k] \\
&= (k + 1)a^k(a - b)
\end{aligned}$$

Thus, we have shown that when the claim is assumed true for $k \geq 1$, then this implies that it is also true for $k+1$. Thus, by the Principle of Mathematical Induction, we have shown the result. □

3. (8 points) Strong Induction Proofs

(a) Show that for all $n \in \mathbb{Z}^+$, n can be written as the sum of distinct powers of 2. (Consider two cases for your inductive step: one where $k+1$ is odd and one where $k+1$ is even. Remember, when $k+1$ is even, then $(k+1)/2$ is an integer).

Proof. **Base Case:** $n = 1$**.** Then, $1 = 2^0$.

Inductive Hypothesis: Assume that for every number $k \geq 1$, we can express k as the sum of distinct powers of 2. Now consider $k+1$.

Case: $k+1$ **is even.** Then, we know that $k+1/2$ is an integer. Note that $k+1/2 \leq k$, so we know by our inductive hypothesis that $k+1/2$ can be written as the sum of direct powers of 2, say $(k+1/2) = 2^{p_1}+2^{p_2}+\cdots+2^{p_i}$, where $p_1 < p_2 < \cdots < p_i$. Then, we have that $k+1 = 2(2^{p_1} + 2^{p_2} + \cdots + 2^{p_i}) = 2^{p_1+1} + 2^{p_2+1} + \cdots + 2^{p_i+1}$, and we maintain the relationship that $p_1 + 1 < p_2 + 1 < \cdots < p_i + 1$, so all of these powers are distinct.

Case: $k+1$ **is odd.** Then, we know that k is even and is also the sum of distinct powers of 2, say $k = 2^{q_1} + 2^{q_2} + \cdots + 2^{q_j}$ where $q_1 < q_2 < \cdots < q_j$. Now note that since k is even, $k/2$ is an integer that is also the sum of distinct powers of 2, namely: $k/2 = 2^{q_1-1} + 2^{q_2-1} + \cdots + 2^{q_j-1}$. Thus, if $q_1 = 0$, we have that $1/2$ is part of the sum of $k/2$. But we know $k/2$ is an integer. Thus, $q_1 > 0$. Note also that $1 = 2^0$. So we have that $k+1 = 2^{q_1} + 2^{q_2} + \cdots + 2^{q_j} + 2^0$, and all such powers are distinct since $0 < q_1 < q_2 < \cdots < q_j$.

Thus, we have shown that when the claim is assumed true for all $k \geq 1$, this implies that the claim is also true for $k+1$. By the Principle of Mathematical Induction, we have shown that the claim is true for all $n \in \mathbb{Z}^+$. □

(b) Prove that if $n \in \mathbb{Z}^+$ such that $n \geq 18$, then n can be expressed as $n = 4x + 7y$, where $x, y \in \mathbb{Z}^+ \cup \{0\}$ (Note: you will need $n = 18, 19, 20, 21$ for your base cases).

Proof. **Base Case 1:** $n = 18$**.** Then, $18 = 14 + 4 = 7 * 2 + 4 * 1$.

Base Case 2: $n = 19$**.** Then, $19 = 12 + 7 = 4 * 3 + 7 * 1$.

Base Case 3: $n = 20$**.** Then, $20 = 4 * 5 + 7 * 0$.

Base Case 4: $n = 21$**.** Then, $21 = 4 * 0 + 7 * 3$.

Inductive Hypothesis: Assume that for every number $k \geq 21$, k can be expressed as $k = 4x + 7y$. Then, for $k+1$, we have that $k+1 = (k-3)+1$. Note that by our inductive hypothesis and our base cases, we know that $k - 3 = 4x' + 7y'$. So then we have that $k = 4x' + 7y' + 3$ and thus

$k+1 = 4x' + 7y' + 3 + 1 = 4x' + 7y' + 4 = 4(x'+1) + 7y'$. Thus, we have expressed $k+1$ as the sum of multiples of 4 and 7 as desired.

Since we have shown that when our claim is true for all $k \geq 21$, it implies that $k+1$ is also true, by the Principle Of Mathematical Induction, the claim is true for all $n \geq 18, n \in \mathbb{Z}$. □

4. (10 points) Recursive Definitions

(a) Let $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$ be the Fibonacci numbers, where F_n is the n^{th} Fibonacci number defined when $n \geq 2$. Prove that for any $n \in \mathbb{Z}^+$, $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$.

Proof. **Base Case: $n = 1$.** Then, $F_{n+1} = F_2 = 1$, $F_{n-1} = F_0 = 0$, and $F_n = F_1 = 1$. Then, $F_{n+1}F_{n-1} - F_n^2 = 1 \cdot 0 - 1^2 = -1 = (-1)^1$. So the claim is true for $n = 1$.

Inductive Hypothesis: Assume that for $k \geq 1$, $F_{k+1}F_{k-1} - F_k^2 = (-1)^k$ is true. Then, consider $k+1$. We have:

$$\begin{aligned}
F_{k+2}F_k - F_{k+1}^2 &= (F_{k+1} + F_k)F_k - (F_k + F_{k-1})^2 \\
&= F_{k+1}F_k + F_k^2 - F_k^2 - 2F_kF_{k-1} - F_{k-1}^2 \\
&= F_{k+1}F_k - 2F_kF_{k-1} - F_{k-1}^2 \\
&= (F_k + F_{k-1})F_k - 2F_kF_{k-1} - F_{k-1}^2 \\
&= F_k^2 + F_kF_{k-1} - 2F_kF_{k-1} - F_{k-1}^2 \\
&= F_k^2 - F_kF_{k-1} - F_{k-1}^2 \\
&= F_k^2 - F_{k-1}(F_k + F_{k-1}) \\
&= F_k^2 - F_{k-1}F_{k+1} \\
&= (-1)(F_{k+1}F_{k-1} - F_k^2) \\
&= (-1)(-1)^k \\
&= (-1)^{k+1}
\end{aligned}$$

We have shown that when the claim is true for k, then it implies that the claim is true for $k+1$ as well. Thus, by the Principle of Mathematical Induction, our claim holds true for all $n \in \mathbb{Z}^+$. □

(b) Suppose you are given n real numbers $a_1, \ldots, a_n$ and you want to find the maximum of those numbers. Give a recursive definition for the function $f(a_1, \ldots, a_n) = \max\{a_1, \ldots, a_n\}$ and prove that this function indeed returns the maximum of these n real numbers.

Solution: Define $f(a_1, \ldots, a_n) = \max\{a_n, f(a_1, \ldots, a_{n-1})\}$, and where the base case is when one element is passed to f: $f(a_1) = a_1$. I claim this function works.

Proof. Suppose we are given n real numbers $a_1, \ldots, a_n$.

Base Case: $n = 1$**.** Then, we only have a_1, and the maximum of one item is the item itself. Similarly, we have that $f(a_1) = a_1$, so the base case holds.

Inductive Hypothesis: Assume that for $k \geq 1$, $f(a_1, \ldots, a_k)$ returns the maximum value of the numbers $a_1, \ldots, a_k$. Then,

$$\begin{aligned} f(a_1, \ldots, a_{k+1}) &= \max\{a_{k+1}, f(a_1, \ldots, a_k)\} && \text{[by definition]} \\ &= \max\{a_{k+1}, a^*\} && \text{[where } a^* = f(a_1, \ldots, a_k)\text{]} \end{aligned}$$

Then note that $f(a_1, \ldots, a_{k+1})$ will return the max of a_{k+1} and a^*, where a^* is the maximum of the values $a_1, dots, a_k$ by our inductive hypothesis. So $f(a_1, \ldots, a_{k+1})$ returns the maximum value of $a_1, \ldots, a_{k+1}$ as desired. By the Principle of Mathematical Induction, we have shown that our definition for f will return the maximum value of n real numbers for $n \geq 1, n \in \mathbb{Z}$. □

(c) (2 points) Let $S = \{(a, b) \colon (a, b) \in \mathbb{Z}^+ \times \mathbb{Z}^+ \text{ and } a + b \text{ is odd}\}$. Give a recursive definition for this set.

Solution: We'll consider two sets such that their union is the set S. Let $(x_0^{(0)}, x_0^{(1)}) = (2, 1)$. Then, define:

$$\begin{aligned} S_x = \{(x_i^{(0)}, x_j^{(1)}) \colon\ & i, j \in \mathbb{Z}^+ \cup \{0\} \\ & \forall i \geq 1, j \text{ is fixed}, (x_i^{(0)}, x_j^{(1)}) = (x_{i-1}^{(0)} + 2, x_j^{(1)}) \\ & \forall j \geq 1, i \text{ is fixed}, (x_i^{(0)}, x_j^{(1)}) = (x_i^{(0)}, x_{j-1}^{(1)} + 2)\} \end{aligned}$$

Similarly, let $(y_0^{(0)}, y_0^{(1)}) = (1, 2)$ and define the set:

$$\begin{aligned} S_x = \{(y_i^{(0)}, y_j^{(1)}) \colon\ & i, j \in \mathbb{Z}^+ \cup \{0\} \\ & \forall i \geq 1, j \text{ is fixed}, (y_i^{(0)}, y_j^{(1)}) = (y_{i-1}^{(0)} + 2, y_j^{(1)}) \\ & \forall j \geq 1, i \text{ is fixed}, (y_i^{(0)}, y_j^{(1)}) = (y_i^{(0)}, y_{j-1}^{(1)} + 2)\} \end{aligned}$$

Then, $S = S_x \cup S_y$.

5. (8 points) Recursive Algorithms

(a) Given $x, n, m \in \mathbb{Z}^+$, write a recursive algorithm to compute $x^n \mod m$ given the fact that $x^n \mod m = (x^{n-1} \mod m \cdot x \mod m) \mod m$. Prove that your algorithm is correct.

Solution: The following algorithm will give the desired outcome:

```
proc MOD(x,n,m)
    if (n == 0)
```

```
        return 1
    z = (x mod m)
    y = MOD(x, (n-1), m)
    return (y*z mod m)
```

Now we prove this algorithm is correct:

Proof. **Base Case:** $n = 0$ We know that any number x to the power 0 is equal to 1 for any modulus m. Thus, in `MOD(x,n,m)` when $n = 0$, the algorithm returns 1 as desired.

Inductive Hypothesis: Assume that our algorithm correctly computes $x^k \mod m$ for $k \geq 0$. Then, call `MOD(x,(k+1),m)`. We know that since we call the algorithm on $k+1$, it will not be in the base case of the algorithm. So the algorithm computes `z = x mod m` and `y = MOD(x,k,m)`. By our inductive hypothesis, we know that $y = x^k \mod m$. Then, our algorithm computes and returns $y \cdot z \mod m = (x^k \mod m \cdot x \mod m) \mod m = x^{k+1} \mod m$ by our fact. Thus, the algorithm correctly returns $x^{k+1} \mod m$.

By the Principle of Mathematical Induction, the algorithm correctly computes $x^n \mod m$ for all $n \geq 0$. □

(b) Give an $\mathcal{O}(n)$-time recursive algorithm to compute the n^{th} Fibonacci number.

Solution: We create a recursive algorithm that returns two values, the n^{th} Fibonacci number F_n and the $(n-1)^{th}$ Fibonacci number F_{n-1}.

```
proc FIB(n):
    if(n == 0)
        return (0,0)
    if(n == 1)
        return (1,0)
    (a,b) = FIB(n-1)
    return (a + b, a)
```

Note that in the line `(a,b) = FIB(n-1)`, $a = F_{n-1}$, $b = F_{n-2}$ and we know that $F_{n-1} + F_{n-2} = F_n$, so we return $(a + b = F_n, a = F_{n-1})$ This function is $\mathcal{O}(n)$ since when not in a base case, we make one recursive call and decrement n by 1, so there will be $n-1$ recursive calls when $n \geq 2$, so this is $\mathcal{O}(n)$.

6. (6 points) Program Correctness

(a) (2 points) Verify that the program segment

`if` $x < 0$ `then` $x := 0$

is correct with respect to the initial assertion **True** and the final assertion $x \geq 0$.

Solution: When the initial assertion is **True** and $x < 0$. Thus, x gets assigned the value 0. Then, the final assertion of $x \geq 0$ is also true since $x = 0$. Now, when the initial assertion is **True** and $x < 0$ is false, this tells us that $x \geq 0$ is true. So again, our final assertion is true. Thus, this program segment is correct with respect to the given initial and final assertions.

(b) Develop rules of inference for the verification of partial correctness of the following program:

```
if  x < 0  then
      y := −2|x|/x
else if  x > 0  then
      y := 2|x|/x
else if  x = 0  then
      y := 2
```

then verify that the program is correct with respect to the initial assertion **True** and the final assertion $y = 2$.

Solution:

Let $y := -2|x|/x$ be the statement S_1, $y := 2|x|/x$ be the statement S_2 and $y := 2$ be the statement S_3. Let C_1 be the condition $x < 0$, let C_2 be the condition $x > 0$, and let C_3 be the condition $x = 0$. Let p be the initial assertion and q the final assertion. Then, our rules of inference with respect to this program segment become:

$$\frac{\begin{array}{l}(p \wedge C_1)\{S_1\}q \\ (p \wedge \neg C_1 \wedge C_2)\{S_2\}q \\ (p \wedge \neg C_1 \wedge \neg C_2 \wedge C_3)\{S_3\}q\end{array}}{\therefore p(\textbf{if } C_1 \textbf{ then } S_1;\ \textbf{else if } C_2 \textbf{ then } S_2;\ \textbf{else if } C_3 \textbf{ then } S_3)q}$$

Now we have the initial assertion **True**. If C_1 is true, then y gets assigned the value $-2|x|/x$. Since C_1 is true, x is negative, so $y = 2$ in this case, and thus the final assertion is correct in this case. If the initial assertion is **True** and C_1 is false and C_2 is true, then y is assigned the value $2|x|/x$. Since C_2 is true, x is positive, so y again simply equals 2. Thus, the final assertion holds true again. Finally, if the initial assertion is true, C_1 is false, C_2 is false, and C_3 is true, we have that y is assigned the value 2. So again the final assertion holds true. Thus, this program segment is correct.

7. (10 points) Counting

(a) (2 points) Let $S = \{1, 2, \ldots, 100\}$. How many subsets of S have exactly 2 elements?

Solution: This is exactly the number of ways to choose 2 elements from this set. Thus, we have that this number is equal to $\binom{100}{2} = 4950$.

(b) How many bit strings of length 10 contain either 5 consecutive 0s or 5 consecutive 1s?

Solution: We first consider only 5 consecutive 0s in our bit string of length 10. Consider first strings of the form:

$$\underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{*}\ \underline{*}\ \underline{*}\ \underline{*}\ \underline{*}$$

Where $*$ can be 0 or 1. Then, the number of strings of this form is exactly $2^5 = 32$.
Now consider strings of the form:

$$\underline{*}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{*}\ \underline{*}\ \underline{*}\ \underline{*}$$

Now before we start counting strings of this form, notice that if the first $*$ is 0, then we have a string that was counted in the first set of strings we considered. So in this case, the strings must look like:

$$\underline{1}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{*}\ \underline{*}\ \underline{*}\ \underline{*}$$

And there are exactly $2^4 = 16$ strings of this form.
Similarly, again shift our five 0s over:

$$\underline{*}\ \underline{*}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{*}\ \underline{*}\ \underline{*}$$

But again, we must be careful not to count strings we've already counted. So these strings must have the form:

$$\underline{*}\ \underline{1}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{*}\ \underline{*}\ \underline{*}$$

This again gives us $2^4 = 16$ such strings.
Continuing:

$$\underline{*}\ \underline{*}\ \underline{1}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{*}\ \underline{*}$$

there are again $2^4 = 16$ of these strings,

$$\underline{*}\ \underline{*}\ \underline{*}\ \underline{1}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{*}$$

$2^4 = 16$ of these strings, and finally

$$\underline{*}\ \underline{*}\ \underline{*}\ \underline{*}\ \underline{1}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}\ \underline{0}$$

$2^4 = 16$ of these strings as well. Thus, we have a total number of strings with five consecutive 0s: $32 + 16 + 16 + 16 + 16 + 16 = 112$.

Note also that the case for five consecutive 1s is identical (swap 1s and 0s in the above argument), so the number of strings with five consecutive 1s is also 112.

Since we wanted to know the total number strings of length 10 with five consecutive 0s or 5 consecutive 1s, we know now that the total number is $112 * 2 = 224$. However, note that in the first and last cases, we've double counted the strings 0000011111 and 0000011111. So we've over-counted by 2. So the total number is $224 - 2 = 222$.

(c) How many functions are there from the set $\{1, 2, \ldots, n\}$ ($n \geq 4$) to the set $\{0, 1, 2\}$ that are

i. one-to-one?

ii. assign exactly 3 numbers less than n to 0?

Solution:

i. Recall that a function f is one-to-one if for any two a, b in the domain, if $f(a) = f(b)$ then $a = b$. Note also that for f to be a function, every element of the domain must map to some element of the codomain. Thus, consider $f\colon \{1, 2, \ldots, n\} \to \{0, 1, 2\}$. Note that since $n \geq 4$, we have a set of size at least 4 mapping into a set of size 3. By the Pigeon Hole Principle, there must be at least 2 distinct elements of the domain that map to the same element in the codomain. Thus, there are no one-to-one functions from $\{1, 2, \ldots, n\}$ to $\{0, 1, 2\}$ when $n \geq 4$.

ii. First, we need to figure out how many subsets of $\{1, 2, \ldots, n-1\}$ of size 3 there are. This is exactly $\binom{n-1}{3}$. Then, once a subset of size 3 is chosen, we have to map every other element for our function. There are exactly $n-3$ elements left to map, and they all have to map to either 1 or 2. Note that it does not matter which they map to. Since we have $n-3$ elements left and exactly two choices of a mapping for each of these elements, we have that the number of mappings for these $n-3$ numbers is exactly 2^{n-3}. Thus, the total number of functions is $\binom{n-1}{3}2^{n-3}$.