College of Engineering & Applied Sciences

CSPB 2824

Discrete Sturctures

Induction Mastery Workbook

Taylor Larrechea

2023

Sections

Mastery Workbook 8

Induction Mastery Workbook

I have neither given nor received unauthorized assistance.

Taylor James Larrechea

Problem 1

Problem Statement

With paper and pencil (or online tablet but not typing), add the numbers 1 - 100 any way you like. Do not research. Maybe look for patterns. Just do it. Describe whatever strategy you used. Include a photo in hw PDF.

Solution

The sum of numbers 1 to 100 is

$$\begin{aligned}
1+2+3+4+5+6+7+8+9+10 &= 55\\
11+12+13+14+15+16+17+18+19+20 &= 155\\
21+22+23+24+25+26+27+28+29+30 &= 255\\
31+32+33+34+35+36+37+38+39+40 &= 355\\
41+42+43+44+45+46+47+48+49+50 &= 455\\
51+52+53+54+55+56+57+58+59+60 &= 555\\
61+62+63+64+65+66+67+68+69+70 &= 655\\
71+72+73+74+75+76+77+78+79+80 &= 755\\
81+82+83+84+85+86+87+88+89+90 &= 855\\
91+92+93+94+95+96+97+98+99+100 &= 955\\
55+155+255+355+455+555+655+755+855+955 &= 5,050
\end{aligned}$$

From the above we can then say

$$\sum_{n=1}^{100}(n) = 5,050. \tag{1}$$

We can verify this result by using the following formula

$$\frac{n(n+1)}{2}. \tag{2}$$

Applying (2) with $n = 100$ we then have

$$\frac{100(100+1)}{2} = \frac{10,100}{2} = 5,050. \tag{3}$$

$1+2+3+4+5+6+7+8+9+10 = \underline{55}$

$11+12+13+14+15+16+17+18+19+20 = \underline{155}$

$21+22+23+24+25+26+27+28+29+30 = \underline{255}$

$31+32+33+34+35+36+37+38+39+40 = \underline{355}$

$41+42+43+44+45+46+47+48+49+50 = \underline{455}$

$51+52+53+54+55+56+57+58+59+60 = \underline{555}$

$61+62+63+64+65+66+67+68+69+70 = \underline{655}$

$71+72+73+74+75+76+77+78+79+80 = \underline{755}$

$81+82+83+84+85+86+87+88+89+90 = \underline{855}$

$91+92+93+94+95+96+97+98+99+100 = \underline{955}$

$55+155+255+355+455+555+655+755+855+955 = \underline{5050}$

$$\sum_{i=1}^{100} i = 5050$$

Just to verify, $n=100$

$$\frac{n(n+1)}{2} = \frac{100(101)}{2} = \frac{10,100}{2} = 5,050$$

Problem 1 Summary

Procedure

- Create a sum for all ten increments of 100.
- Sum up all ten increments to get a final value.
- Compare the result with the formula

$$\sum_{i=1}^{n} = \frac{n(n+1)}{2}.$$

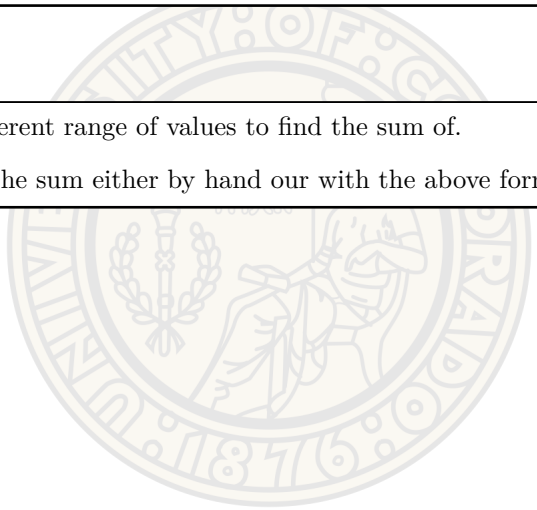
Key Concepts

- This problem encapsulates the counting of numbers from 1 to a given integer.
- We can compare the result from our method by hand with

$$\sum_{i=1}^{n} = \frac{n(n+1)}{2}.$$

Variations

- We could be given a different range of values to find the sum of.
  - We could perform the sum either by hand our with the above formula.



Problem 2

Problem Statement

Understanding summation notation is essential in computer science. There will be an exam question about summation notation. Review/practice the exercises #29, #31, #32 on page 169 in Rosen as needed. You may do these in the Workbook but only select the graded portion below to submit. Check your answers using Math Is Fun.

Pick any ONE example from 29, 31, 32 and do the following on a single page.

- Write the summation example chosen in summation notation.
- Write out the terms and solution by hand.
- Write, in text, a short python function to calculate this sum without using built in functions for sums.
- Answer "How are summations and For Loops related?"

Solution - #29

For this problem, I am going to do problem 29 from Rosen on page 169. Particularly part (d) of 29.

29. What are the values of these sums?

(d) $\sum_{j=0}^{8}(2^{j+1} - 2^j)$

$$\sum_{j=0}^{8}(2^{j+1} - 2^j) = (2^1 - 2^0) + (2^2 - 2^1) + (2^3 - 2^2) + (2^4 - 2^3) + (2^5 - 2^4) + (2^6 - 2^5) + (2^7 - 2^6) + (2^8 - 2^7) + (2^9 - 2^8)$$

$$= (1) + (2) + (4) + (8) + (16) + (32) + (64) + (128) + (256) = 511.$$

The code for this summation in Python can be seen below.

```python
# Part d
def PartD(j0,jn):
    result = 0
    for i in range(j0,jn + 1):
        result += (2**(i + 1) - 2**i)
    return result

```

```
# Function call with print statement
print(f"The result from part (d) of problem 29 is: {PartD(0,8)}")
# Output
The result from part (d) of problem 29 is: 511

```

Summations and for loops are related in that they are being executed over a range of integers. In a summation, we are performing some sort of mathematical operation for a given index in the range and then summing those results over the entire range. For loops do something similar but they aren't constrained to performing mathematical operations inside of them. They can be used for a myriad of different operations with summing being one of them.

Problem 2 Summary

Procedure

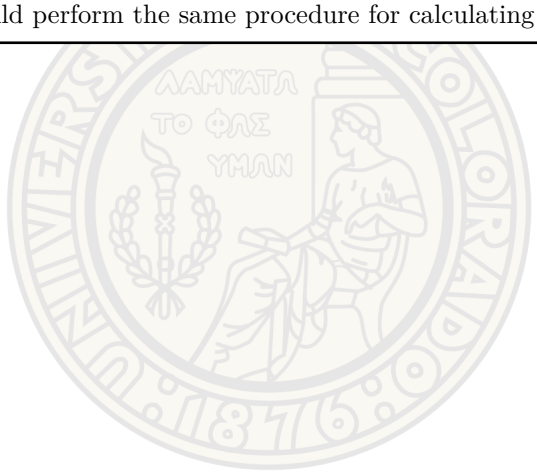
- Perform the sum for the expression $\sum_{j=0}^{8}(2^{j+1} - 2^{j})$ by hand.
- Write code in Python for a generalized range of this sum.
- Display the result and generate of the summary the problem.

Key Concepts

- This problem depicts the use of summation notation and how it is applied for a specific expression.
- We can generalize this sum by writing a code in Python that will perform the sum.
- Summations and for loops are related in that they both perform operations over a given range.

Variations

- We could be given a different expression.
  - In this case we would perform the same procedure for calculating the sum.



Problem 3

Problem Statement

Do #3, #9 and #13 as needed page 168 for practice. Do #12 b) and c) with explanation on this page.

Solution - #12

12. Show that the sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = -3a_{n-1} + 4a_{n-2}$ if

(b) $a_n = 1$

$$\begin{aligned} -3a_{n-1} + 4a_{n-2} &= -3(1) + 4(1) && \text{(Substitution)} \\ &= -3 + 4 && \text{(Simplification)} \\ &= 1 && \text{(Addition)} \\ &= a_n && \text{(Substitution)} \end{aligned}$$

(c) $a_n = (-4)^n$

$$\begin{aligned} -3a_{n-1} + 4a_{n-2} &= -3(-4)^{n-1} + 4(-4)^{n-2} && \text{(Substitution)} \\ &= -3(-4)^n(-4)^{-1} + 4(-4)^n(-4)^{-2} && \text{(Algebra)} \\ &= -3(-4)^n(-4)^{-1} + 4(-4)^n(-4)^{-1}(-4)^{-1} && \text{(Algebra)} \\ &= (-4)^n(-3(-4)^{-1} + 4(-4)^{-1}(-4)^{-1}) && \text{(Factoring)} \\ &= (-4)^n\left(\frac{-3}{-4} + \frac{4}{(-4)\cdot(-4)}\right) && \text{(Simplification)} \\ &= (-4)^n\left(\frac{3}{4} + \frac{4}{16}\right) && \text{(Simplification)} \\ &= (-4)^n\left(\frac{3}{4} + \frac{1}{4}\right) && \text{(Simplification)} \\ &= (-4)^n(1) && \text{(Addition)} \\ &= (-4)^n && \text{(Simplification)} \\ &= a_n && \text{(Substitution)} \end{aligned}$$

When showing that a sequence is a solution to a recurrence relation we first need to substitute in the provided sequence into the recurrence relation. After this substitution we then need to carry out the algebra and show that the resulting algebra will result in the provided sequence.

For part (b) of this problem, the sequence that was provided to us was a constant. In this case, the terms a_{n-1} and a_{n-2} have no effect on the constant term and we show that resulting algebra will be equal to the sequence that was provided to us. In part (c), when we insert $a_n = (-4)^n$ into the recurrence relation we have to apply the terms a_{n-1} and a_{n-2} to the provided sequence. When we do this and carry out the algebra we get back the sequence that was provided to us. These relations always end with the LHS of the recurrence relation being equal to a_n.

Problem 3 Summary

Procedure

- Start with the recurrence relation and substitute the recurrence relation into it.
- Perform the necessary algebra to express the substituted sequence in its simplest form.
- Show that the simplest form is equal to the recurrence relation.

Key Concepts

- When showing that a sequence is a solution for a recurrence relation, we first substitute the sequence into the recurrence relation.
- For sequences that are constants, it does not matter what the recurrence relation is, the sequences will be unchanged.
- The methodology for showing that a sequence is a solution to a recurrence relation is showing that once substituted, the algebra will simplify to the original recurrence relation.

Variations

- We could be given a different recurrence relation and sequence.
  - We would then perform the same procedure of substituting the sequence into the recurrence relation and then going through the necessary algebra that is needed to arrive at the original recurrence relation.

Problem 4

Problem Statement

For all positive integers $n, 1 * 2 + 2 * 3 + \cdots + n(n+1) = n(n+1)(n+2)/3$.

Solution

Theorem: For all positive integers, the sum of the products of n consecutive integers is one third the product of $n(n+1)(n+2)$. Namely,

$$\forall n \in \mathbb{Z}^+, \sum_{i=1}^{n}(i(i+1)) = \frac{n(n+1)(n+2)}{3}.$$

Proof: By mathematical induction.

Base Case: With $P(n) = \frac{n(n+1)(n+2)}{3}$, $P(1)$ is

$$P(1) = \frac{1(1+1)(1+2)}{3} = \frac{1(2)(3)}{3} = \frac{6}{3} = 2 = 1 \cdot 2.$$

Inductive Hypothesis: There is an arbitrary positive integer k such that

$$1 \cdot 2 + 2 \cdot 3 + \cdots + k(k+1) = \frac{k(k+1)(k+2)}{3}.$$

Inductive Step: If

$$1 \cdot 2 + 2 \cdot 3 + \cdots + k(k+1) = \frac{k(k+1)(k+2)}{3}.$$

then

$$1 \cdot 2 + 2 \cdot 3 + \cdots + k(k+1) + (k+1)(k+2) = \frac{(k+1)(k+1+1)(k+2+1)}{3} = \frac{(k+1)(k+2)(k+3)}{3}.$$

Induction Proof:

$$\begin{aligned}
1 \cdot 2 + 2 \cdot 3 + \cdots + k(k+1) + (k+1)(k+2) &= \frac{k(k+1)(k+2)}{3} + (k+1)(k+2) && \text{(Adding } (k+1)(k+2) \text{ To Each Side)} \\
&= \frac{k(k+1)(k+2)}{3} + \frac{3(k+1)(k+2)}{3} && \text{(Algebra)} \\
&= \frac{k(k+1)(k+2) + 3(k+1)(k+2)}{3} && \text{(Adding Fractions)} \\
&= \frac{(k+1)(k+2)(k+3)}{3} && \text{(Factoring } (k+1)(k+2)\text{)}
\end{aligned}$$

Consequently, we thus have $P(1)$ and $\forall k \in \mathbb{Z}^+, P(k) \rightarrow P(k+1)$, so by the principle of mathematical induction, it follows that $P(n)$ is true for all positive integers n. $\square$

Problem 4 Summary

Procedure

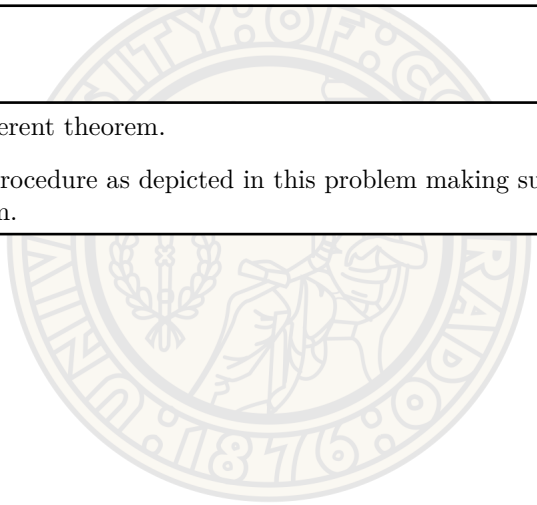
- Begin by stating the original theorem.
- State that the proof is going to be done with the method of mathematical induction.
- Calculate the base case of the theorem.
- State the inductive hypothesis (substitute k for the other variable in the expression).
- State the inductive step (substitute $k+1$ for every k in the inductive hypothesis).
- Start with the induction hypothesis, transform it into the inductive step, and proceed to prove 'then' part of the inductive step.
- Conclude the proof with showing that $P(\text{base case}) \rightarrow P(\text{inductive step})$.

Key Concepts

- This problem showcases the method of a proof by induction.

Variations

- We could be given a different theorem.
  - Perform the same procedure as depicted in this problem making sure to use the correct base case for the new theorem.



Problem 5

Problem Statement

For all positive integers $n, 1^3 + 2^3 + 3^3 + 4^3 + \cdots + n^3 = (n(n+1)/2)^2$.

Solution

Theorem: For all positive integers, the sum of the cube of integers is the square of one half of the product of $n(n+1)$. Namely,

$$\forall n \in \mathbb{Z}^+, \sum_{i=1}^{n}(i^3) = \left(\frac{n(n+1)}{2}\right)^2.$$

Proof: By mathematical induction.

Base Case: With $P(n) = \left(\frac{n(n+1)}{2}\right)^2$, $P(1)$ is

$$P(1) = \left(\frac{1(1+1)}{2}\right)^2 = \left(\frac{(1)(2)}{2}\right)^2 = (1)^2 = 1 = 1^3.$$

Inductive Hypothesis: There is an arbitrary positive integer k such that

$$1^3 + 2^3 + 3^3 + 4^3 + \cdots + k^3 = \left(\frac{k(k+1)}{2}\right)^2.$$

Inductive Step: If

$$1^3 + 2^3 + 3^3 + 4^3 + \cdots + k^3 = \left(\frac{k(k+1)}{2}\right)^2.$$

then

$$1^3 + 2^3 + 3^3 + 4^3 + \cdots + k^3 + (k+1)^3 = \left(\frac{(k+1)(k+1+1)}{2}\right)^2 = \left(\frac{(k+1)(k+2)}{2}\right)^2.$$

Induction Proof:

$$\begin{aligned}
1^3 + 2^3 + 3^3 + 4^3 + \cdots + k^3 + (k+1)^3 &= \left(\frac{k(k+1)}{2}\right)^2 + (k+1)^3 && \text{(Adding } (k+1)^3 \text{ To Each Side)}\\
&= \frac{k^2(k+1)^2}{4} + (k+1)^3 && \text{(Carrying Out Square Of First Term)}\\
&= \frac{k^2(k+1)^2}{4} + \frac{4(k+1)^3}{4} && \text{(Algebra)}\\
&= \frac{k^2(k+1)^2 + 4(k+1)^3}{4} && \text{(Adding Fractions)}\\
&= \frac{(k+1)^2(k^2 + 4(k+1))}{4} && \text{(Factoring } (k+1)^2\text{)}\\
&= \frac{(k+1)^2(k^2 + 4k + 4)}{4} && \text{(Distribution)}\\
&= \frac{(k+1)^2(k+2)(k+2)}{4} && \text{(Polynomial Factorization)}\\
&= \frac{(k+1)^2(k+2)^2}{4} && \text{(Grouping Like Terms)}\\
&= \left(\frac{(k+1)(k+2)}{2}\right)^2 && \text{(Simplification)}
\end{aligned}$$

Consequently, we thus have $P(1)$ and $\forall k \in \mathbb{Z}^+, P(k) \rightarrow P(k+1)$, so by the principle of mathematical induction, it follows that $P(n)$ is true for all positive integers n. $\square$

Problem 5 Summary

Procedure

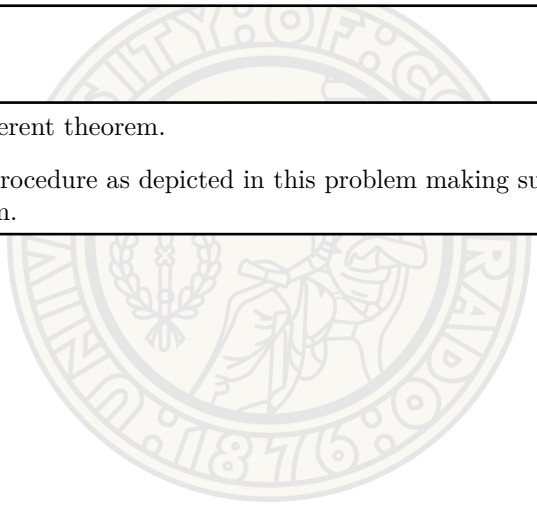
- Begin by stating the original theorem.
- State that the proof is going to be done with the method of mathematical induction.
- Calculate the base case of the theorem.
- State the inductive hypothesis (substitute k for the other variable in the expression).
- State the inductive step (substitute $k+1$ for every k in the inductive hypothesis).
- Start with the induction hypothesis, transform it into the inductive step, and proceed to prove 'then' part of the inductive step.
- Conclude the proof with showing that $P(\text{base case}) \rightarrow P(\text{inductive step})$.

Key Concepts

- This problem showcases the method of a proof by induction.

Variations

- We could be given a different theorem.
  - Perform the same procedure as depicted in this problem making sure to use the correct base case for the new theorem.



Problem 6

Problem Statement

You may use PythonTutor or Runestone's code lens to help you understand. Do #3 (Ungraded on your own). Then answer here #4 from page 370, by hand, show work. Provide brief insight/explanation/notes about how this is different and/or the same as your RSA code.

Solution - #4

4. Trace Algorithm 3 when it finds gcd(12, 17). That is, show all the steps used by Algorithm 3 to find gcd(12, 17).

Algorithm 3:

procedure gcd(a, b: nonnegative integers with $a < b$)
if $a = 0$ then return b
else return gcd(b **mod** a, a)
{output is gcd(a, b)}

Trace Of Algorithm 3:

We are aiming to calculate the gcd(12, 17). In each execution of this recursive algorithm we will denote the updated value for a as a' and b as b'.

Initial Iteration:	$\gcd(12, 17)$	$a = 12, b = 17$
	$a \neq 0 \therefore \gcd(17 \textbf{ mod } 12 = 5, 12)$	$a' = 5, b' = 12$
Second Iteration:	$\gcd(5, 12)$	$a = 5, b = 12$
	$a \neq 0 \therefore \gcd(12 \textbf{ mod } 5 = 2, 5)$	$a' = 2, b' = 5$
Third Iteration:	$\gcd(2, 5)$	$a = 2, b = 5$
	$a \neq 0 \therefore \gcd(5 \textbf{ mod } 2 = 1, 2)$	$a' = 1, b' = 2$
Fourth Iteration:	$\gcd(1, 2)$	$a = 1, b = 2$
	$a \neq 0 \therefore \gcd(2 \textbf{ mod } 1 = 0, 1)$	$a' = 0, b' = 1$
Fifth Iteration:	$\gcd(0, 1)$	$a = 0, b = 1$
	$a = 0 \therefore$ return b, $b = 1$	

Following the procedure of algorithm 3 we see that gcd(12,17) = 1.

Synopsis:

Algorithm 3 is utilizing recursion to calculate the gcd of two integers. This algorithm is immediately different than that of the algorithm in our RSA project because it utilizes recursion. Our algorithm in RSA because it is utilizing a while loop inside the algorithm (at least in my case) until a certain condition is met for a value that is calculated inside the loop. The EEA algorithm in our project is also returning the Bezout coefficients for two integers and algorithm 3 is only returning the gcd of two integers.

Problem 6 Summary

Procedure

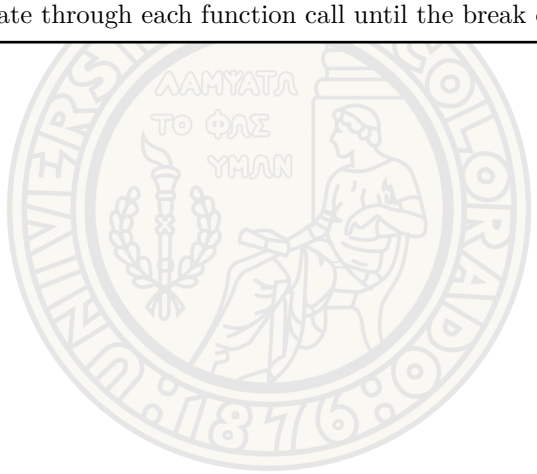
- Iterate through the recursive algorithm by performing the operation that is defined in the algorithm.
- Update the input parameters from the return values that are returned from one iteration until the break condition is reached.
- Show what the final return value of the algorithm will be after the break condition has been reached.

Key Concepts

- This problem showcases the method of how a recursive algorithm is performed.
- Each iteration of a recursive algorithm will return values that are then fed into the algorithm again until the break condition is reached.

Variations

- We could be given a different algorithm to analyze for what the algorithm will return.
  - We would then iterate through each function call until the break condition has been reached.



Problem 7

Problem Statement

Do #5 (Ungraded on your own). Do and show #6 page 370, by hand, show work. Provide brief insight/explanation/notes about how this is different and/or the same as your RSA code.

Solution - #6

6. Trace Algorithm 4 when it is given $m = 7, n = 10$, and $b = 2$ as input. That is, show all the steps Algorithm 4 uses to find 2^{10} **mod** 7.

Algorithm 4:

procedure mpower(b, n, m: integers with $b > 0$ and $m \geq 2, n \geq 0$)
if $n = 0$ then
  return 1
else if n is even then
  return mpower$(b, n/2, m)^2$ **mod** m
else
  return (mpower$(b, \lfloor n/2 \rfloor, m)^2$ **mod** $m \cdot b$ **mod** m) **mod** m
{output is b^n **mod** m}

We are aiming to calculate b^n **mod** m. In each execution of this recursive algorithm we will denote the updated value for b as b', n as n', and m as m'.

Initial Iteration:	mpower(2,10,7)	$b = 2, n = 10, m = 7$
	$n \neq 0, n$ is even $\therefore$ return mpower$(2, 5, 7)^2$ **mod** 7	$b = 2, n' = 5, m = 7$
Second Iteration:	mpower(2,5,7)	$b = 2, n = 5, m = 7$
	$n \neq 0, n$ is odd $\therefore$ return (mpower$(2, 2, 7)^2$ **mod** $7 \cdot 2$) **mod** 7	$b = 2, n' = 2, m = 7$
Third Iteration:	mpower(2,2,7)	$b = 2, n = 2, m = 7$
	$n \neq 0, n$ is even $\therefore$ return mpower$(2, 1, 7)^2$ **mod** 7	$b = 2, n' = 1, m = 7$
Fourth Iteration:	mpower(2,1,7)	$b = 2, n = 1, m = 7$
	$n \neq 0, n$ is odd $\therefore$ return (mpower$(2, 0, 7)^2$ **mod** $7 \cdot 2$) **mod** 7	$b = 2, n' = 0, m = 7$
Fifth Iteration:	mpower(2,0,7)	$b = 2, n = 0, m = 7$
	$n = 0 \therefore$ return 1	

Now using the results from all of the recursive calls we have

$$\begin{aligned}
\text{Fourth Iteration Result} &= (\text{mpower}(2,0,7)^2 \cdot 2) \textbf{ mod } 7 = (1^2 \textbf{ mod } 7 \cdot 2) \textbf{ mod } 7 = (1 \cdot 2) \textbf{ mod } 7 = 2 \textbf{ mod } 7 = 2 \\
&\therefore \text{mpower}(2,1,7) = 2 \\
\text{Third Iteration Result} &= (\text{mpower}(2,1,7)^2) \textbf{ mod } 7 = 2^2 \textbf{ mod } 7 = 4 \textbf{ mod } 7 = 4 \\
&\therefore \text{mpower}(2,2,7) = 4 \\
\text{Second Iteration Result} &= (\text{mpower}(2,2,7)^2 \cdot 2) \textbf{ mod } 7 = (4^2 \textbf{ mod } 7 \cdot 2) \textbf{ mod } 7 = (2 \cdot 2) \textbf{ mod } 7 = 4 \textbf{ mod } 7 = 4 \\
&\therefore \text{mpower}(2,5,7) = 4 \\
\text{Initial Iteration Result} &= (\text{mpower}(2,5,7)^2) \textbf{ mod } 7 = 4^2 \textbf{ mod } 7 = 16 \textbf{ mod } 7 = 2 \\
&\therefore \text{mpower}(2,10,7) = 2
\end{aligned}$$

Climbing back up the recursive tree we can see that 2^{10} **mod** $7 = 2$.

Synopsis:

This algorithm that we used in this problem is different from our FME algorithm in the RSA project as this algorithm is using recursion to calculate the FME of integers. In our RSA project we used a for loop that iterated over the digits in a binary number and performed a specific operation with that digit depending on its value. In my opinion, this algorithm is a lot more difficult to interpret because of the recursive calls. The fact that this recursive algorithm in particular returns a value with the function call again makes it more difficult to interpret.

Problem 7 Summary

Procedure

- Iterate through the recursive algorithm by performing the operation that is defined in the algorithm.
- Update the input parameters from the return values that are returned from one iteration until the break condition is reached.
- Climb back up the recursive tree to show what each branch of the tree will evaluate to.
- Show what the final branch of the tree will evaluate to at the end of the recursive calls.

Key Concepts

- This problem showcases the method of how a recursive algorithm is performed.
- Each iteration of a recursive algorithm will return values that are then fed into the algorithm again until the break condition is reached.
- We can use results from branches that are farther from the root of the tree to incur what the initial call of the algorithm will be.

Variations

- We could be given a different algorithm to analyze for what the algorithm will return.
  - We would then iterate through each function call until the break condition has been reached.
  - We then would use results from the last call to incur what the previous calls evaluate to.