Week 6 Modular Arithmetic Continued Lecture Note

Notebook:	Computational Mathematics		
Created:	2020-04-21 2:48 PM	**Updated:**	2020-05-07 5:45 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** Modular Arithmetic Continued	Course: BSc Computer Science Class: Computational Mathematics[Lecture] Date: May 07, 2020

Essential Question:

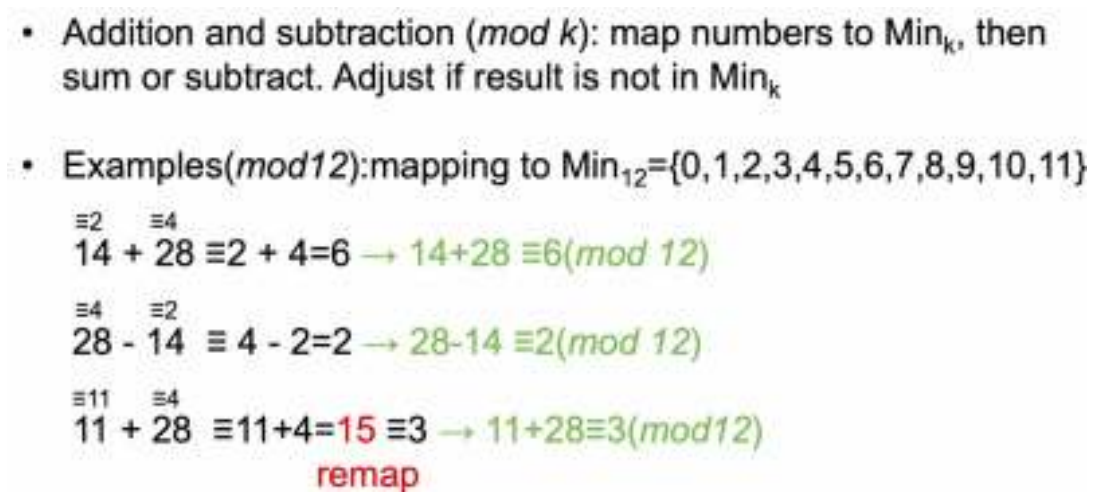
What are the operations that we can perform in modular arithmetic?

Questions/Cues:

- How do we perform addition and/or subtraction in modular arithmetic?
- How do we perform multiplication in modular arithmetic?
- How do we perform division in modular arithmetic?

Notes

- Addition/Subtraction(in mod) = map the operands(numbers) being added or subtracted to the minimal subset of the mod term, remembering that the minimal subset is 0 to k-1. Mapping the numbers to the subset simply mean that each number is replaced by its equivalent congruency from minimal subset. Remember that this equivalent congruency is obtained by dividing each number being operated on by the mod term ie. 12 and comparing the remainder to see which number in the minimal subset each number is congruent to and replacing it with such and performing the said operation, if after the operation the resultant number is not in the subset, divide the number by the mod term until the number is one of the ones in the minimal subset



- Multiplication(in mod) = map the operands(numbers) being multiplied to the minimal subset of the mod term, remembering that the minimal subset is 0 to k-1. Mapping the numbers to the subset simply mean that each number is replaced by its equivalent congruency from minimal subset. Remember that this equivalent congruency is obtained by dividing each number being operated on by the mod term ie. 12 and comparing the remainder to see which number in the minimal subset each number is congruent to and replacing it with such and performing the multiplication, if after the multiplication the resultant number is not in the subset, divide the number by the mod term until the number is one of the ones in the minimal subset

- Multiplication (*mod k*): map numbers to Min_k, then multiply. Adjust if result is not in Min_k

- Examples(*mod 12*): mapping to {0,1,2,3,4,5,6,7,8,9,10,11}

$14 \times 28 \overset{\equiv 2}{\equiv} 2 \times 4 \overset{\equiv 4}{=} 8 \rightarrow 14 \times 28 \equiv 8 (mod\ 12)$

$11 \times 28 \overset{\equiv 11}{\equiv} 11 \times 4 = 44 \overset{\equiv 4}{\equiv} 8 \rightarrow 11 \times 28 \equiv 8 (mod\ 12)$

- Division(in mod) = Division in modular arithmetic can be straightforwardly extended and applied like the previous operations mentioned. When mapping a "divisor/divided" to the minimal subset, you must make sure that the dividend is defined during congruency because division by zero is not defined. To extend division, you must first define the multiplicative inverse in modulus k, the multiplicative inverse m^{-1} of the integer m is such that m times m^{-1} = 1 in modulus k. Then, we can define the division of two numbers in modulus k as $a \times b^{-1} (\mathrm{mod}\ k)$

- Division cannot be straightforwardly extended
- Example (*mod6*) 4/12 not defined since $12 \equiv 0$ (*mod 6*)
- Define first multiplicative inverse (*mod k*)
- Multiplicative inverse m^{-1} of integer m : $m \times m^{-1} = 1 (mod\ k)$
- Then define a/b (*mod k*) as $a \times b^{-1}$ adopting multiplication
- Example: find 2^{-1} (*mod 7*)? $\rightarrow Min_7 = \{0,1,2,3,4,5,6\}$

Try all integers in Min_7

$2 \times 0 \equiv 0 (mod\ 7)$ $2 \times 1 \equiv 2 (mod\ 7)$ $2 \times 2 \equiv 4 (mod\ 7)$ $2 \times 3 \equiv 6 (mod\ 7)$
$2 \times 4 \equiv 1 (mod\ 7)$ $2 \times 5 \equiv 3 (mod\ 7)$ $2 \times 6 \equiv 6 (mod\ 7)$

$\rightarrow 2^{-1} = 4 (mod\ 7)$

- Division Example: $6 / 2 (\mathrm{mod}\ 7) = 6 \times 2^{-1} = 6 \times 4 = 24 \equiv 3 (\mathrm{mod}\ 7)$

- *Note: inverse cannot always be found*

  Ex. : inverse of 2(*mod4*) Min_4={0,1,2,3}

  $2\times0\equiv0\ (mod\ 4)$ $2\times1=2\equiv2$ $2\times2=4\equiv0$ $2\times3=6\equiv2$

Summary

In this week, we learned about the different operations we can perform in modular arithmetic.