Data Analysis and Algorithm

Practical 10

Implement Chinese reminder theorem to a constraint satisfaction problem. Analyze its complexity..

Date.: 24-10-21

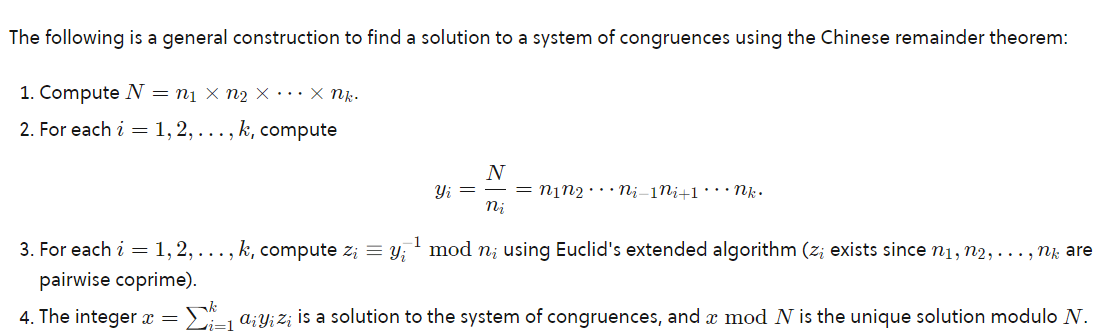
Name – Yash Vasudeo Prajapati

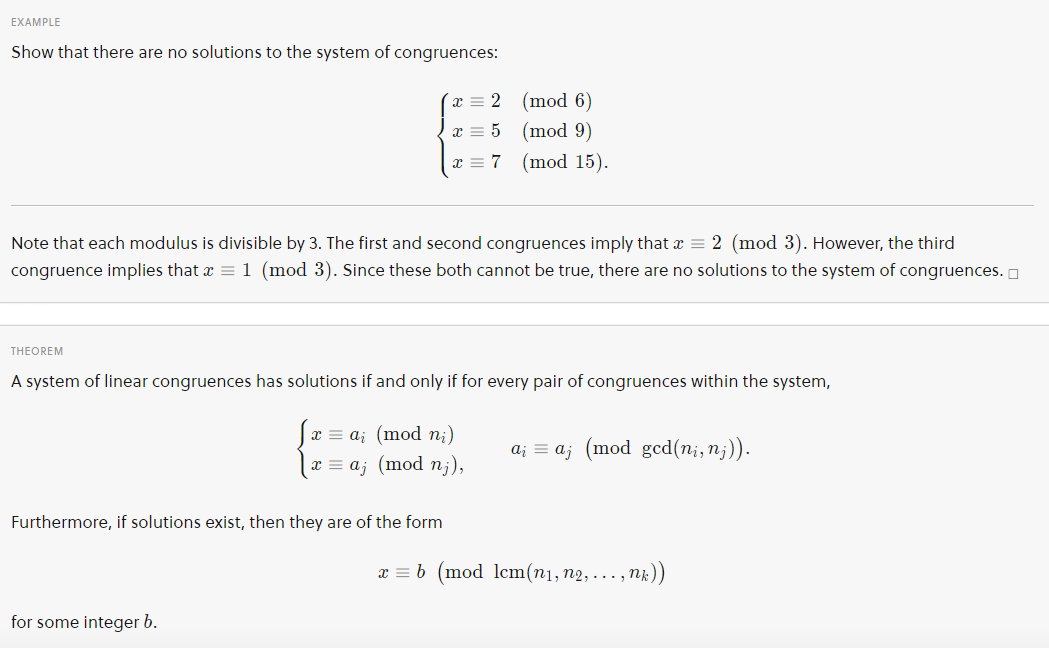
Rollno - 022

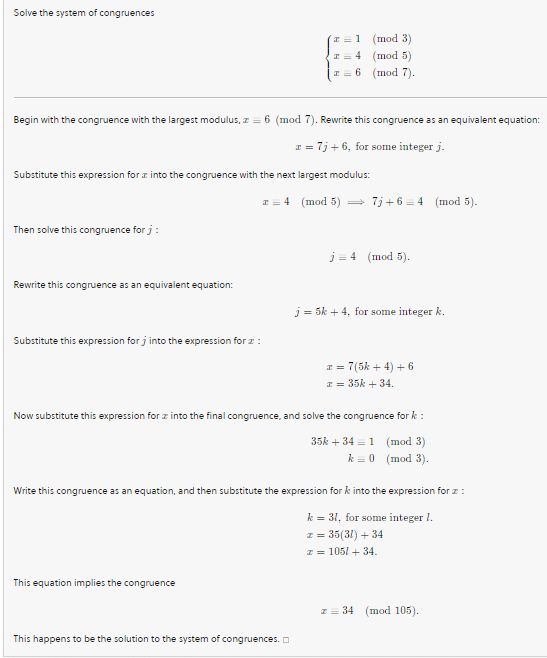
MSc. Computer Science

Theory:-

In number theory, the Chinese remainder theorem states that if one knows the remainders of the Euclidean division of an integer n by several integers, then one can determine uniquely the remainder of the division of n by the product of these integers, under the condition that the divisors are pairwise coprime.





****

Program:-

Program 1

def reminder(a, b):

b0 = b

x0, x1 = 0, 1

if b == 1: return 1

while a > 1:

q = a // b #35 // 3

a, b = b, a%b #a = 3, b = 2

#x0 = 1- (11 \*0), x1=1

x0, x1 = x1 - q \* x0, x0

if x1 < 0:

x1 += b0

return x1

if \_\_name\_\_ == '\_\_main\_\_':

from functools import reduce

a = [2, 3, 2]

#a(mod n)

n = [3, 5, 7]

#return the N = n1\*n2\*n3

t = reduce(lambda a, b: a\*b, n)

Ni = [ t//i for i in n]

xi = [ reminder(Ni[i], n[i]) for i in range(0,len(a))]

total = [a[i]\*Ni[i]\*xi[i] for i in range(0,len(a))]

total = sum(total)

x = total % t

print(x)

Program 2

from functools import reduce

def chinese\_remainder(n, a):

sum = 0

#return the N = n1\*n2\*n3

prod = reduce(lambda a, b: a\*b, n)

#zip give {n1:a1,n2:a2...}

for ni, ai in zip(n, a):

p = prod // ni

print(ni)

sum += ai \* mul\_inv(p, ni) \* p

return sum % prod

def mul\_inv(a, b):

b0 = b

x0, x1 = 0, 1

#if ni = 1

if b == 1: return 1

#else p > 1

while a > 1:

q = a // b #35 // 3

a, b = b, a%b #a = 3, b = 2

x0, x1 = x1 - q \* x0, x0

if x1 < 0:

x1 += b0

return x1

a = [2, 3, 2]

n = [3, 5, 7]

print(chinese\_remainder(n,a))

Output:-



Complexity Analysis:-

The complexity of Chinese reminder theorem is O(n) where n is the number of input / size of array .

The program only needs one loop to calculate the product of all the n.

Conclusion:-

We implement Chinese reminder theorem and find that its complexity is O(n).