DESIGN ANALYSIS AND ALGORITHM

PRACTICAL NO 10

027_Abhishek_Ojha

Experiment No – 10 **Date of Experiment:- 06th December 2021**

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

Algorithm for Chinese Remainder Theorem

• **Step 1:-** Find M=m, * m, * ... * m. This is the

common modulus.

Step 2:- Find My = M/m, M, = M/m2, ...,Mk =

M/m.

• **Step 3:-** Find the multiplicative inverse of M1,

M2, ...,Mk using the corresponding moduli (m,

m2,

m). Call the inverses as:-

M, ?,M,',..., M.: -1

• **Step 4:** The solution to the simultaneous

equations is:-

x = (a, XM, XM, ' + a2 XM, XM, ' + ... +ak * My x MA-") mod M

Input: num[] = {5, 7}, rem[] = {1, 3}

Output: 31

Explanation:

31 is the smallest number such that:

(1) When we divide it by 5, we get remainder 1.

(2) When we divide it by 7, we get remainder 3.

Input: num[] = {3, 4, 5}, rem[] = {2, 3, 1}

Output: 11

Explanation:

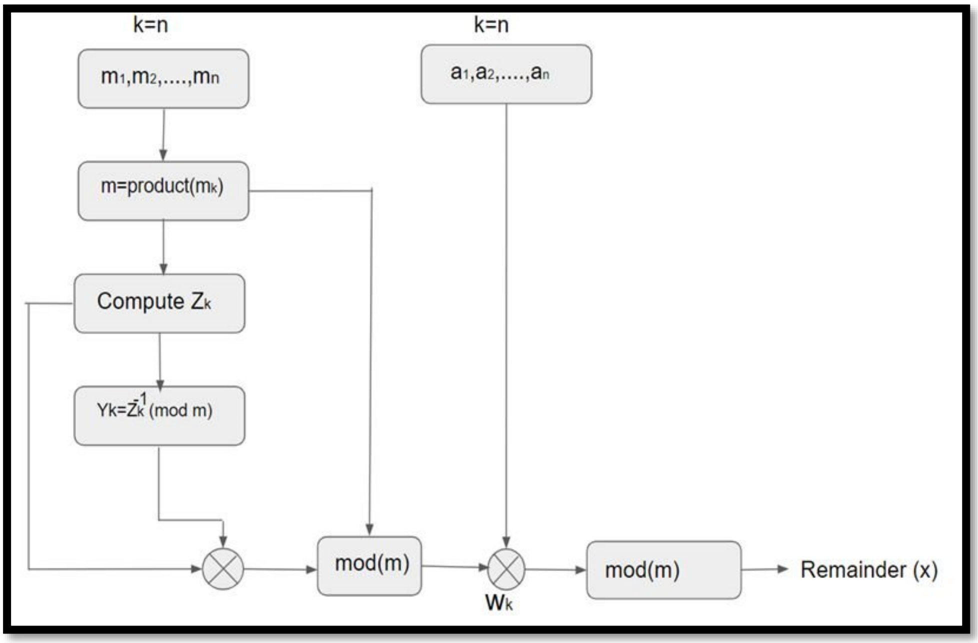
11 is the smallest number such that:

(1) When we divide it by 3, we get remainder 2.

(2) When we divide it by 4, we get remainder 3.

(3) When we divide it by 5, we get remainder 1.

Figure:



Practical Implementation of Chinese Remainder Theorem

```python
# Chinese Remainder Theorem
def findMinX(num, rem, k):
    x = 1;
    while(True):

        j = 0;
        while(j < k):
            if (x % num[j] != rem[j]):
                break;
            j += 1;
        if (j == k):
            return x;
        x += 1;
num = [3, 4, 5];
rem = [2, 3, 1];
k = len(num);
print("x is", findMinX(num, rem, k));
```

Output:

```
x is 11
```

Time Complexity: O(M), M is the product of all elements of num[] array.

Auxiliary Space: O(1)

Conclusion: Successfully Implemented the Chinese Remainder Theorem.