Discrete Maths

Assignment 3

Seif Mohamed Mahmoud Gneedy
18010834

Ans Gomaa Elnagar
18010421

➢ Problem statement

Q1 : Implementing sieve of Eratosthenes algorithm for finding all prime numbers up to any given limit.

Q2: Implementing Trial Division algorithm for integer factorization to check that a given number is prime or not.

Q3: Implementing extended Euclidean algorithm to find the greatest common divisor ‘d’ of two positive integers ‘a’ and ‘b’.
In addition, it outputs Bezout’s coefficients ‘s’ and ‘t’ such that “d = s*a + t*b”.

Q4: implementing Chinese remainder theorem that takes as input “m1,m2,...,mn” that are pairwise reltively prime and “a1,a2,...,an” and calculates ‘x’ such that :
x=a1(mod m1)
x=a2(mod m2)
.
.
x=an(mod mn)

Q5: implementing Miller’s test to check if a certain number is probably prime or not prime.

➢ Used Data structure

Sieve of Eratosthenes : Used ArrayList of Integers to store all primes in it.

Trial Division : No special DS.

Extended Euclidean Algorithm : an Array contain 3 integers :{ GCD , S , T } returns as result from the method.

Chinese Remainder Theorem : Used ArrayList to store m and a lists and also MK’s and inverse of MK’s mod m’s .

Miller's Test : No special DS.

➢ **Pseudo code**

Sieve of Eratosthenes : Getting all prime numbers under a given number.

```
method sieveOfEratosthenes(Integer number){
    Creating an ArrayList to store result in it
    Adding numbers (from 2 to number) to the result list
    iterate over every element in the list{
        removing elements that is divisible by current element
        except current one
    }
    now the list have just the primes then,
    return result list
}
```

Trial Division : Checking if the given number is prime or not.

```
method trialDivision(Integer number){
    Iterate with i starting from 2 till square root of number{
        check if number is divisible by i :
                return false
    }
    return true as There's no divisors except 1
}
```

Extended Euclidean Algorithm : Finding GCD and Bezout's coefficients and returning the result as array contains{GCD,S,T}.

```
method extendedEuclideanAlgorithm(Integer a,Integer b){
    defining initial values of s0,s1,t0,t1
    Iterate till remainder equals zero{
        getting quotient of a/b operation
        calculating s(j)=s(j-2)-q(j-1)*s(j-1) and updating s0,s1
        calculating t(j)=t(j-2)-q(j-1)*t(j-1) and updating t0,t1
        calculating remainder and store it in b variable
```

updating a
}
Creating array to store result in { a , s0 , t0 } as it represents {GCD,S,T}
return result array
}

Chinese Remainder Theorem : Calculates the value of x from the given System of linear Congruences with “m1,..,mn” are pairwise relative prime.

method chineseRemainderTheorem(List a,List m){
defining a variable to store the product of m’s (mProduct)
iterate over m’s{
store the product of m’s in mProduct
}
Creating ArrayList M (the product of m’s except m(K))
iterate over m’s with i variable {
M(i)=mProduct/m(i)
}
Creating ArrayList inverseList (inverse of M(k) mod m(k)
iterate with i from 0 to end of m’s list{
getting the inverse of M(k) mod m(k) by extended Euclidean Algorithm defined above.
check if negative then add m(k)
store it in inverse list
}
getting the result by calculating
*result = a(k)*M(k)*inverseList(k) as k=0,..nOfM’s*
return result mod mProduct
}

Miller's Test : Test if a given number is not prime or probably prime.

method millerTest(Integer number){
returning false if number is even or 1
handling case if n<=3
*getting an odd number such that number-1 = (2^k)*r*
So defining d=number -1
iterating till d become odd with updating d with every iteration.
iterating 10 times to get more accurate test{
getting random number in range [2,...,number-1]
store it in a variable
compute x = (a^d)%number
iterate till d becomes number-1 again {
check if x reached number-1 then return true
check if x equals 0 then return false
updating x =(x^2)%number
*updating d=d*2*
}

}
return false
}

➢ Sample runs

Sieve of Eratosthenes :

```
Choose method :
1) sieve of Eratosthenes.
2) Trial Division.
3) extended Euclidean Algorithm.
4) Chinese remainder theorem.
5) Miller's test
6) Exit.
Enter choice :1

Enter the required limit:100
The primes till 100 are :
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

Trial Division :

```
Choose method :
1) sieve of Eratosthenes.
2) Trial Division.
3) extended Euclidean Algorithm.
4) Chinese remainder theorem.
5) Miller's test
6) Exit.
Enter choice :2
Enter the number :101
The number 101 is prime.
```

Extended Euclidean Algorithm :

```
Choose method :
1) sieve of Eratosthenes.
2) Trial Division.
3) extended Euclidean Algorithm.
4) Chinese remainder theorem.
5) Miller's test
6) Exit.
Enter choice :3
Enter the first number :252
Enter the second number :198
GCD is :18
Bezout's coefficients are : s = 4 t = -5
```

Chinese Remainder Theorem :

```
Welcome to assignment 3
Choose method :
1) sieve of Eratosthenes.
2) Trial Division.
3) extended Euclidean Algorithm.
4) Chinese remainder theorem.
5) Miller's test
6) Exit.
Enter choice :4
As equations has the form :
        X=a1(mod m1) ... X=an(mod mn).
Enter input in this form : (a1,m1),(a2,m2),..,(an,mn)
(2,3),(3,5),(2,7)
The solution is : 23
```

Miller’s Test :

```
Choose method :
1) sieve of Eratosthenes.
2) Trial Division.
3) extended Euclidean Algorithm.
4) Chinese remainder theorem.
5) Miller's test
6) Exit.
Enter choice :5
Enter The number you want to test :1093
The number 1093 is probably prime.
```