Cryptography Assignment - 2

Pramod Aravind Byakod

1. Compute gcd(269, 35) and find its representation (i.e. x and y such that 269x+35y=gcd(269,35)). Show step-by-step calculations.

269 = 7*35 + 24
35 = 1*24 + 11
24 = 2*11 +2
11 = 5*2 + 1 $\rightarrow$ GCD
2 = 2*1 + 0

Let's represent it in the form 269x+35y = 1
Referring to above equations, we could write
1 = 11 – 5*2
1 = 11 – 5*(24 – 2*11)
1 = 11 – 5*24 + 10*11
1 = 11*11 – 5*24
1 = 11*(35 – 1*24) – 5*24
1 = 11*35 – 11*24 - 5*24
1 = 11*35 – 16*24
1 = 11*35 – 16*(269 – 7*35)
1 = 11*35 – 16*269 + 112*35
1 = 269*(-16) + 35*123

Therefore, x =-16 and y=123

2. Consider the following Sage/Python program to compute the gcd of two integers a and b (assume that a > b > 0):

```
for i in range(b,0,-1):
  if a % i ==0 and b % i ==0:
    print i
    break
```

Is the algorithm correct? Is it efficient? Justify your answer

Algorithm is correct, it finds the GCD of two positive integers a and b, where a is greater than b.

Coming to efficiency, let's take few examples and analyze the code.

Example 1, take a = 9 and b = 5

Iteration -1. i = 5, 9%5 != 0

Iteration -2 i = 4, 9%4 != 0

Iteration -3 i = 3, 9%3 = 0 and 5%3!=0

Iteration -4 i = 2, 9%2 != 0

Iteration -5 i = 1, 9%1 = 0 and 5%1 = 0, print 1, break

Example 2, take a = 4 and b = 2

Iteration -1 i = 2, 4%2 = 0 and 2%2 = 0, print 2, break

Observing both cases, number of iterations increases as the value of b increases. Example 2 gives the best-case condition where the GCD is found in the first iteration. Whereas example 2 gives the worst-case condition, where the iterations are equal to the value of b. Consider

two prime numbers, a=101 and b=97. Here, the for loop is ran 97 times to find the GCD, since both the numbers are primes and their GCD is 1.

In conclusion, the above algorithm may give the correct output but fails in efficiency, as the time complexity is too high.

3. Consider the following Sage/Python program to compute the gcd of two integers a and b (assume that a > b > 0):

```
while b != 0:
  a = a -b
  if a<b:
    a, b= b,a
print a
```

Is the algorithm correct? Is it efficient? Justify your answer

Algorithm is correct, it finds the GCD of two positive integers a and b, where a is greater than b.

Will consider the examples as in the question 2

Example 1, a = 9,b = 5

Iteration 1 – a=4, 4<5, a=5 , b=4

Iteration 2 – a=1, 1<4, a=4 , b=1

Iteration 3 – a=3, 3!<1, b=1

Iteration 4 – a=2, 2!<1, b=1

Iteration 5 – a=1, 1!<1, b=1

Iteration 6 – a=0, 0<5, a=1, b=0, print 1

Example 2, a = 4,b = 2

Iteration 1 – a=2, 2!<2, b=2

Iteration 2 – a=0, 0<2, a=2, b=0, print 2

For example 1, program took 6 iterations and for example 2 it took 2 iterations.

Example 1 is the worst-case condition and example 2 is one of the best-case conditions. Comparing to the algorithm in the question 2, this algorithm down performs. Because, for the same examples, algorithm 1 took 5 iterations whereas algorithm 2 took 6 iterations. In case of example 2 also, algorithm 2 took one iteration more than the algorithm 1.

In conclusion, by the above analysis, this algorithm may give the correct output but fails to perform in an efficient way.

Euclidian algorithm for computing GCD is the most efficient algorithm. The Euclidean algorithm is based on the principle that the greatest common divisor of two numbers does not change if the larger number is replaced by its difference with the smaller number. Since this replacement reduces the larger of the two numbers, repeating this process gives successively smaller pairs of numbers until the two numbers become equal.