Advanced Algorithms

Floriano Zini

Free University of Bozen-Bolzano
Faculty of Computer Science

Academic Year 2013-2014

Lab 3 – Exercises on algorithms with numbers

Exercise 1.12 page 39 DPV

- What is 2^(2^2006) (mod 3)?

Exercise 1.12 page 39 DPV

- What is 2^(2^2006) (mod 3)?

- Solution
  - $2^{(2^{2006})} = 2^{(2\cdot 2^{2005})} = 4^{(2^{2005})} \equiv 1 \bmod 3$

Exercise 1.23 page 40 DPV

- Show that if a has a multiplicative inverse modulo *N*, then this inverse is unique (modulo *N*)
  - Hint: proof by contradiction assuming there are two distinct inverses

Exercise 1.23 page 40 DPV

- Show that if a has a multiplicative inverse modulo *N*, then this inverse is unique (modulo *N*)

- Solution
  - Suppose x1 and x2 are two distinct inverses of a mod N. Then,

    $$x1=x1*1=x1*a*x2=1*x2=x2 \pmod{N}$$

    which is a contradiction

Exercise 1.27 page 40 DPV

- Consider an RSA key set with $p = 17, q = 23$
  $N = p*q = 391$, and $e = 3$.
  What value of d should be used for the secret key?
  What is the encryption of the message $M = 41$?

Exercise 1.27 page 40 DPV

- Consider an RSA key set with $p = 17, q = 23$
  $N = p*q = 391$, and $e = 3$.
  What value of d should be used for the secret key
  What is the encryption of the message $M = 41$?

- Solution
  - First calculate $(p-1)*(q-1) = 16 * 22 = 352$.
    We use the extended Euclid algorithm to compute the $gcd(3,352)$ and get the inverse d of $e \bmod 352$. We easily obtain $e*d \equiv 1 \bmod 352 \rightarrow d \equiv -117 \equiv 235 \bmod 352$.
    The encryption of the message $M=41$ is
    $E(M)=M^e \bmod N = 41^3=117*41 = 105 \bmod 391$

Exercise

```
function extended-Euclid(a, b)
Input: Two positive integers a and b with a ≥ b ≥ 0    a ≠ 0
Output: Integers x, y, d such that d = gcd(a, b) and ax + by = d

if b = 0: return (1, 0, a)
(x', y', d) = extended-Euclid(b, a mod b)
return (y', x' − ⌊a/b⌋y', d)
```

- Implement in Octave the extended Euclid's algorithm

Exercise

```
function extended-Euclid(a, b)
Input: Two positive integers a and b with a ≥ b ≥ 0    a ≠ 0
Output: Integers x, y, d such that d = gcd(a, b) and ax + by = d

if b = 0: return (1, 0, a)
(x', y', d) = extended-Euclid(b, a mod b)
return (y', x' − ⌊a/b⌋y', d)
```

- Implement in Octave the extended Euclid's algorithm
- Solution

```
function g = gdc_ext(a,b)
if (b == 0)
   g = [1 0 a];
else
   gp=gcd_ext(b,mod(a,b));
   g=[gp(2) gp(1)-floor(a/b)*gp(2) gp(3)];
endif;
end
```

Assignment 02

Exercise 1.28 page 40 DPV

- In an RSA cryptosystem, $p = 7$ and $q = 11$. Find appropriate exponents d and e.

Assignment 02 (cont.)

Exercise 1.33 page 41 DPV

- Give an efficient algorithm to compute the *least common multiple* of two n-bit numbers x and y, that is, the smallest number divisible by both x and y. What is the running time of your algorithm as a function of n?

Assignment 02 (cont.)

Exercise

- Implement in Octave the algorithm you have found in the previous exercise to calculate the least common multiple.