MATH 314 Spring 2020 - Class Notes

02/12/2019

Scribe: Luke Dixon

Summary: Today's class included continued discussion and examples of Euclid's Algorithm, the Extended Euclidean Algorithm, Modular Exponetiation and Fermat's Little Theorem.

Notes:

Euclidean Algorithm continued

- compute gcd(72,26) = gcd (26,20)
- Division with remainder = gcd(20,6)
- 72 =(26) +20
- 26 = 1(20)+6
- 20 = 3(6)+ 2
- 6 = 3(2) + 0
- Running time of Euclid's Algorithm is O($\log^2(a+b)$).

Extended Euclidean Algorithm

Theorem: If gcd(a,b) = d, then by using Euclid's Algorithm backward we can find two integers m and n such that ma + nb = d - linear combination of a and b.

2 = 20-3(6)
6 = 26 -1(20)
2 = 20-3(26-1(20))
2 = -3(26) + 4(20)
20 = 72-2 (26)
2 =-3(26)+4(72-2(26))
2 = 4(72) -11(26)
If gcd(a,b)=1 then there exist m and n so that ma + nb = 1
reduce this (mod b)
ma + 0 = 1 (mod b)
ma = 1 (mod b)
How to find inverse a -1(mod n)
Use Euclid's Algorithm to find gcd(a,n)
If this isn't 1, give up!
Use Euclid's Algorithm backward

to find k,l so that ka+ ln =1
reduce (mod n)
ka = 1 (mod n)
so k = a inverse -1 (mod n)
Example: solve 27x + 3 = 10 (mod 50)
27x = 7 (mod 50)
We need inverse 27 (mod 50)
Apply Euclid's Algorithm gcd(50,27)
50 = 1(27) + 23
27 = 1(23) + 4
23 = 5(4) + 3
4 = 1(3) + 1
3 = 3(1) +0
so inverse 27 = 13 (mod 50)
1 = 4-1(3)
3 = 23-5(4)
1 = 4-1(23-5(4))
= -1(23)+6(4)
4 = 27 -1(23)
= -1(23)+6(27-1(23))
1 = 6(27) - 7(23)
23 = 50-1(27)
1 = 6(27)-7(50-1(27))
1 = -7(50)+ 13(27)
1 = -7(50) + 13(27) (mod 50)
1 = 13(27)(mod 50)
13(27) = 13*7 (mod 50)
X = 91 = 41 (mod 50)
x = 41 (mod 50)
a = 27 in this case
<u>Modular Exponentiation</u>
compute a$^m (mod n)$
write m in binary as a sum of powers of two.
<u>Repeated Squaring</u>
- square a as many times as digits in binary of m.
- reduce mod n every time and multiply together terms from binary expression for m.
Example: compute 3 $^{71} (mod 11)$
71 = 64 + 4 + 2 + 1
$2^6 + 2^2 + 2^1 + 2^0$
1000111_2