Jared Gridley — Algorithms — HW 2

1.9) Sum:

$x \equiv x' \mod N, \quad y \equiv y' \mod N \Rightarrow x+y \equiv x'+y' \mod N$

This means they will have the same remainder when divided by N

$23 \equiv 9 \mod 7$, $2 \equiv 2 \mod 7$ $\longrightarrow$ $23-9 = 14 = 2N$

$\frac{14}{2} = 7 = N$

Always goes back to a multiple of N.

$x+y \equiv x'+y' \mod N \longrightarrow (x+y)-(x'+y') = cN$

$y \equiv y' \mod N \longrightarrow y-y' = aN$ (a is a constant $\in \mathbb{Z}$)

$x \equiv x' \mod N \longrightarrow x-x' = bN$ (b is a constant $\in \mathbb{Z}$)

$(y-y')+(x-x') = aN+bN$

$(y+x)-(y'+x') = (a+b)N$ ($a+b$ is constant $\in \mathbb{Z}$)

$x+y \equiv x'+y' \mod N$

Since $(a+b)N$ means that the difference will be a multiple of N, then we can conclude that the remainders of the addition will be the same, and thus holds.

Multiplication: $x \equiv x' \mod N, \quad y \equiv y' \mod N \Rightarrow xy \equiv x'y' \mod N$ mod N

$x-x' = aN$, $y-y' = bN$ $\} \rightarrow (x-x')(y-y') = abN^2$ $\longrightarrow$ Need to get $xy-x'y'$

$x = x'+aN$

$y = y'+bN$

$(x-x')(y-y') = abN^2$

$xy-xy'-x'y+x'y' = abN^2$

$xy-(xy'+x'y)+x'y' = abN^2$

$xy-(y'(x'+aN)+x'(y'+bN))+x'y' = abN^2$

Need to get rid of all the remaining x and y because we already have the xy term.

$xy-(y'x'+y'aN+x'y'+x'bN)+x'y' = abN^2$

$xy-x'y'-(aNy'+bNx') = abN^2$

$xy-x'y' = abN^2+(ay'+bx')N$

$xy-x'y' = (abN+ay'+bx')N$

This shows us that the difference will be a multiple of N, so the two remainders must be the same.

$\Rightarrow xy \equiv x'y' \mod N$

1.11) Is $4^{1536} - 9^{4824}$ divisible by 35?

Try to find a simpler number for N than 35.

35 = 7 × 5

5, 7 are both prime, so they will be the smallest factors of 35 to help simplify the problem.

If 4^{1536} and 9^{4824} are both multiples of 5 and 7, then their difference will be a multiple of 35 (as shown by previous problem)

Divisible by 7:

$4^{1536} \bmod 7$

$4^2 = 16$

$4^3 = 64$ r1

$64 \to 1 \bmod 7$

$(4^3)^{512} \to 1^{512}$

$4^{1536} = 1 \bmod 7$

$9^{4824} \bmod 7$

$7 \times 9 = 63$

$7 \times 10 = 70$

$7 \times 11 = 77$

$7 \times 12 = 84$

If we use this one, then we can reuse the work for checking 4^{1536}

$9^2 = 81 = 4 \bmod 7$

$\Rightarrow (4)^{2412} \bmod 7$

$= (4^3)^{804} \bmod 7$

$4^3 = 64 = 1 \bmod 7$

$= 1^{804} = 1 \bmod 7$

$4^{1536} - 9^{4824} = (1 \bmod 7) - (1 \bmod 7) = 1 - 1 \bmod 7$

$= 0 \bmod 7$

So difference is divisible by 7

Divisible by 5:

$4 \times 3 = 12$

$4 \times 4 = 16$

$4 \times 5 = 20$

Cannot be 5 b/c exponent not div by 5 evenly

4^{1536}

$\Rightarrow (4^4)^{384} \bmod 5$

$= 1^{384} \bmod 5 = 1$

$9^{4824} \bmod 5$

$9 \times 9 = 81$

$81 \bmod 5 = 1$

$9^{4824} = (9^2)^{2412} = (81)^{2412}$

$1^{2412} \bmod 5 = 1$

$4^{1536} - 9^{4824} \bmod 5 = 1 \bmod 5 - 1 \bmod 5$

$= 1 - 1 \bmod 5 = 0 \bmod 5$

So the difference is divisible by 5

Since the difference is divisible by both 5 and 7, it must also be divisible by 35.

1.12) What is $2^{2^{2006}} \bmod 3$

$4 \times 5 = 20$
$3 \times 7 = 21$
} Won't work $2006 \% 5 \neq 0$

$2^{2^{2006}} = 4^{2006} \bmod 3$

$4^{2006} = (4^2)^{1003} = 16^{1003}$

$(16 \bmod 3)^{1003} = 1^{1003} = 1$

(Tinkering)

$4^2 = 16$

$16 \bmod 3 = 1$

$$2^{2^{2006}} \bmod 3 = 4^{2006} \bmod 3 = (4^2 \bmod 3)^{1003}$$

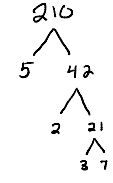
$$= 1^{1003} \bmod 3 = \boxed{1}$$

1.18) Compute $\gcd(210, 588)$

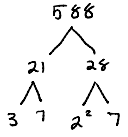
Factorization:

Factors of 210: 2, 3, 5, 6, 7, 10, 14, 15, 21, 30, 35, 42, 70, 105

Factors of 588: 2, 3, 4, 6, 7, 12, 14, 21, 28, 42, 49, 84, 98, 147, 196, 294



$210 = 5 \times 2 \times 3 \times 7$

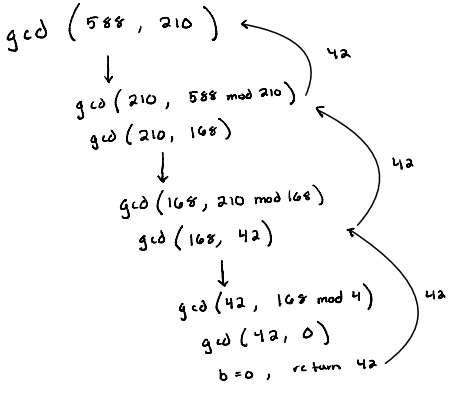


$588 = 3 \times 7^2 \times 2^2$

$7 \times 3 \times 2 = 42$

$\gcd(210, 588) = 42$

Euler's Algorithm:



$\gcd(588, 210) = 42$

1.31) Consider $N! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot N$

a) how many bits long is $N!$ in $\Theta(n)$ form

From Stirling's Approximation, we know that the bit length of a factorial is $O(\log_2(\text{Factorial}))$

$\Longrightarrow \log_2(N!) \cdot \log_2(N \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 2 \cdot 1) = \log_2(N) + \log_2(n-1) + \ldots \log_2(2) + \log_2(1)$

Magnitude of all the terms is relatively close to $\log_2(n)$, so for our approximation we will treat them like they are all the same magnitude of $\log_2(n)$

$\Longrightarrow \Theta(\log_2 n!) = \boxed{\Theta(n \log_2 n)}$

b) Algorithm for computing $N!$ with runtime analysis.

$N! = n(n-1)(n-2)\ldots(1)$ → Because multiplication is communitive, we can use a for loop to multiply From 1 to N

$N! = (1)(2)\ldots(n-1)(N)$

```
NFactorial(N)
    result = 1
    for i in range(1, N+1):
        result = result * i
    return result
```

← Will go through n-iterations

← Multiplication of 2 n-bit numbers will take $O(n^2)$

Bit length $O(n)$ x Bit length $O(n)$

# of iterations $O(n)$ x time for multiplication $O(n^2)$

$\boxed{O(n^3) \text{ time}}$

Lab 2 Results:

- 3 trials of 100, 1000, 10000 digit numbers

Results:		
	Regular	Div and Conq
100	4.29E-06	8.30E-05
	2.62E-06	2.72E-05
	2.86E-06	3.81E-05
Avg	3.26E-06	4.94E-05
1000	3.10E-06	4.60E-05
	2.15E-06	2.88E-05
	2.15E-06	4.10E-05
Avg	2.46E-06	3.86E-05
10000	4.29E-06	4.60E-05
	2.86E-06	4.51E-05
	3.81E-06	4.60E-05
Avg	3.66E-06	4.57E-05