Ilana Sivan
205634272
Honors Course
Homework 2

Question 1

Justify the correctness of the algorithm div(x, y) from class, that is to say show that it returns two values (q, r) st. $\mathrm{x} = \mathrm{yq} + \mathrm{r}$, $\mathrm{r} < \mathrm{y}$, and explain why the runtime is $O(n^2)$.

We prove that for all $x \in \mathbb{N}$, the algorithm will return (q, r) st. $x = qy + r$

$(q'$,$r') = Div(\lfloor \frac{x}{2} \rfloor, y) = Div(\frac{x}{2}, y)$ is recursive.

Let $x = qy + r$, $\lfloor \frac{x}{2} \rfloor = q'y + r'$, and $0 \leq \mathrm{r} \leq \mathrm{y} - 1$, $0 \leq \mathrm{r}' \leq \mathrm{y} - 1$.

We have $r' < y$, and from this it follows that $2r' < 2y$, $r = 2r' < 2y$, so $r - y < y$.

$$\lfloor \tfrac{\mathrm{x}}{2} \rfloor = \begin{cases} \frac{x}{2} & x \rightarrow even \\ \frac{x-1}{2} & x \rightarrow odd \end{cases}$$

$$\mathrm{x} = \begin{cases} 2\lfloor \frac{x}{2} \rfloor = 2q'y + 2r' & x \rightarrow even \\ 2\lfloor \frac{x}{2} \rfloor + 1 = 2q'y + 2r' + 1 & x \rightarrow odd \end{cases}$$

We explain and further continue:
If x is even: $x = 2\lfloor \frac{x}{2} \rfloor = 2q'y + 2r'$, and q = 2q', r = 2r'

If x is odd: $x = 2\lfloor \frac{x}{2} \rfloor + 1 = 2q'y + 2r' + 1$, and $\mathrm{q} = 2\mathrm{q}'$, $\mathrm{r} = 2\mathrm{r}' + 1$

Therefore, we can conclude that $0 \leq \mathrm{r}' \leq \mathrm{y} - 1$ and $1 \leq 2\mathrm{r}' + 1 \leq 2\mathrm{y} - 1$

If $\mathrm{r} \geq \mathrm{y}$:
$0 \leq \mathrm{r} - \mathrm{y} \leq \mathrm{y} - 1$,
$\mathrm{q} = \mathrm{q} + 1$,
$\mathrm{r} = \mathrm{r} - \mathrm{y}$
Our recursion is correct :)

The algorithm terminates after at most n recursive calls, each call halves x (for $n$halves), and we reduce the number of bits by one. Therefore, there will be no more than $n$executions of our algorithm. Each recursive call requires a total of $O(n)$ bit operations, so the total time taken is $O(n^2)$.
Formally:

$T(x, y) = T(\frac{\mathrm{x}}{2}, y) + \Theta(n)$.

As our x is represented in $n-$bits, we can represent moving to the right with $(n-1)$ bits, since moving to the right is an $\frac{x}{2}$ operation. Therefore: $T(n) = T(n-1) + \Theta(n) = T(n-2) + \Theta(n) = ... = T(1) + \Theta(n)$. Our $T(1)$ is a constant, and we have $n$iterations for $T(n) = O(n^2)$

Question 2

Using the definition of $\mathrm{x} \equiv \mathrm{y} \bmod N$ (namely, that N divides $\mathrm{x} - \mathrm{y}$), prove:

$\mathrm{x} \equiv \mathrm{x}' \mathrm{mod} N$,$\mathrm{y} \equiv \mathrm{y}' \mathrm{mod} N \Rightarrow \mathrm{x} + \mathrm{y} \equiv \mathrm{x}' + \mathrm{y}' \bmod N$

$\mathrm{x} \equiv \mathrm{x}' \mathrm{mod} N$,$\mathrm{y} \equiv \mathrm{y}' \mathrm{mod} N \Rightarrow \mathrm{xy} \equiv \mathrm{x}'\mathrm{y}' \bmod N$

Let $\mathrm{x} \mathrm{mod} N = a$, and $\mathrm{y} \mathrm{mod} N = b$.

From this we can see that $x + y \mathrm{mod} N = a + b$.

We can write our a as $a = x' \mathrm{mod} N$, as $x \equiv x' (\mathrm{mod} N)$.

Similarly, we can write b as $b = y' \mathrm{mod} N$, as $y \equiv y' (\mathrm{mod} N)$.

Therefore, we have $(x + y) \mathrm{mod} N = (x' + y') \mathrm{mod} N$.

We can conclude then that $x + y \equiv x' + y' (\mathrm{mod} N)$ Let $\mathrm{x} \mathrm{mod} N = a$, and $\mathrm{y} \mathrm{mod} N = b$.

Now, we see that $x * y (\mathrm{mod} N) = a * b$

We can write our a as $a = x' \mathrm{mod} N$ as $x \equiv x' (\mathrm{mod} N)$.

We write our b as $b = y' \mathrm{mod} N$ as $y \equiv y' (\mathrm{mod} N)$

Therefore we have $(x * y) \mathrm{mod} N = (x' * y') \mathrm{mod} N$

So we conclude, $x * y = x' * y' (\mathrm{mod} N)$

Question 3

Answer the following questions:

a. Is $4^{1536} - 9^{4824}$ divisible by 35? **The answer is yes:**

$4^{1536} \equiv 16^{768} \equiv 256^{384} \equiv 11^{384} \equiv 11^{96} \equiv 11^{24} \equiv 11^{6} \equiv 11^{4} * 11^{2} \equiv 11 * 16 \equiv 1 \mathrm{mod} 35$

$9^{4824} \equiv 81^{2412} \equiv 11^{2412} \equiv 11^{603} \equiv 1331^{201} \equiv 1^{201} \equiv 1 \mathrm{mod} 35$

Note: $11^4 \equiv 11$.

b. What is $2^{2^{2006}} (\text{mod } 3)$?

$2^{2^{2006}} \text{mod} 3 \equiv 2^{2^{2005*2}} \equiv (2^{2^{2005}})^2 \equiv (2^2)^{2^{2005}} \equiv 1^{2^{2005}} \equiv 1 \text{mod} 3$

Therefore, $2^{2^{2006}}$ **mod**$3 = 1$**.**

c. Is the difference of $5^{30,000}$ and $6^{123,456}$ a multiple of 31?

We would like to determine if the difference between 6^{123456} and $5^{30,000}$ is divisible by 31:

$6^{123,456} \equiv 5^{30,000} \text{mod} 31$

$5^{30,000} - 6^{123,456} \equiv (5^6)^{5000} - 6^{123,456} \equiv (15,625)^{5000} - (6^6)^{20,576} \equiv 1^{5000} - 1^{20,576} \equiv 0 \text{mod} 31$

The answer is yes.

Question 4

Choose your favorite programming language and implement:

a. The multiplication algorithm of **Karatsuba** for inputs of arbitrary lengths $\text{n} \in \mathbb{N}$.

b. The exponentiation algorithm we saw in class for input: $\text{x}, \text{y}, N \in \mathbb{N}$, and output: $\text{x}^{\text{y}} \text{mod} N$

I have provided question 4 in separate files. Thank you.