CS 170: Algorithms

Account forms now or after class!

CS 170: Algorithms

Account forms now or after class!

Static Course Webpage. (inst.cs.berkeley.edu/c̃s170)

CS 170: Algorithms

Account forms now or after class!

Static Course Webpage. (inst.cs.berkeley.edu/c̃s170)

Will mostly use piazza. Should have/get invitation soon.

CS 170: Algorithms

Account forms now or after class!

Static Course Webpage. (inst.cs.berkeley.edu/c̃s170)

Will mostly use piazza. Should have/get invitation soon.

Did you find a scanner, yet?

Today.

Modular arithmetic.

Today.

Modular arithmetic.
...up to ...

Today.

Modular arithmetic.
...up to ...

Arithmetic.

Addition: $O(n)$

Arithmetic.

Addition: $O(n)$

$$\begin{array}{cccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ + & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\ \hline \end{array}$$

Arithmetic.

Addition: $O(n)$

$$\begin{array}{cccccccccc}
 & & & & & & & & 1 & \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
+ & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
 & & & & & & & & & 0
\end{array}$$

Arithmetic.

Addition: $O(n)$

$$
\begin{array}{cccccccccc}
 & & & & & & & 1 & 1 & \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
+ & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
 & & & & & & & & 8 & 0
\end{array}
$$

Arithmetic.

Addition: $O(n)$

$$
\begin{array}{cccccccccc}
 & & & & & & 1 & 1 & 1 & \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
+ & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
 & & & & & & & 4 & 8 & 0
\end{array}
$$

Arithmetic.

Addition: $O(n)$

$$
\begin{array}{cccccccccc}
 & & & & & 1 & 1 & 1 & 1 & \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
+ & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
 & & & & & & 4 & 4 & 8 & 0
\end{array}
$$

Arithmetic.

Addition: $O(n)$

$$
\begin{array}{cccccccccc}
 & & & & 0 & 1 & 1 & 1 & 1 & \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
+ & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
 & & & & & 9 & 4 & 4 & 8 & 0
\end{array}
$$

Arithmetic.

Addition: $O(n)$

$$
\begin{array}{cccccccccc}
 & & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
+ & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
 & & & 4 & 6 & 9 & 4 & 4 & 8 & 0
\end{array}
$$

Arithmetic.

Addition: $O(n)$

$$\begin{array}{cccccccccc}
 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
+ & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
 & & 4 & 4 & 6 & 9 & 4 & 4 & 8 & 0
\end{array}$$

Arithmetic.

Addition: $O(n)$

1	0	0	0	0	1	1	1	1	
	1	2	3	4	5	6	7	8	9
+	9	2	1	2	3	7	6	9	1
	0	4	4	6	9	4	4	8	0

Arithmetic.

Addition: $O(n)$

$$
\begin{array}{cccccccccc}
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
+ & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
1 & 0 & 4 & 4 & 6 & 9 & 4 & 4 & 8 & 0
\end{array}
$$

Arithmetic.

Addition: $O(n)$

1	0	0	0	0	1	1	1	1	
	1	2	3	4	5	6	7	8	9
+	9	2	1	2	3	7	6	9	1
1	0	4	4	6	9	4	4	8	0

Time: $O(n)$

Arithmetic.

Addition: $O(n)$

1	0	0	0	0	1	1	1	1	
	1	2	3	4	5	6	7	8	9
+	9	2	1	2	3	7	6	9	1
1	0	4	4	6	9	4	4	8	0

Time: $O(n)$

Can we do better?

Arithmetic.

Addition: $O(n)$

$$
\begin{array}{cccccccccc}
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
+ & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
1 & 0 & 4 & 4 & 6 & 9 & 4 & 4 & 8 & 0
\end{array}
$$

Time: $O(n)$

Can we do better?

Need to look at the numbers to add them...

Arithmetic.

Addition: $O(n)$

1	0	0	0	0	1	1	1	1	
	1	2	3	4	5	6	7	8	9
+	9	2	1	2	3	7	6	9	1
1	0	4	4	6	9	4	4	8	0

Time: $O(n)$

Can we do better?

Need to look at the numbers to add them... optimal.

More Al Khwarizmi's: algorithms.

Addition: $O(n)$

More Al Khwarizmi's: algorithms.

Addition: $O(n)$

Multiplication:

$$
\begin{array}{rrrrrrrrrr}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\times & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
\end{array}
$$

More Al Khwarizmi's: algorithms.

Addition: $O(n)$

Multiplication:

$$
\begin{array}{rccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\times & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9
\end{array}
$$

More Al Khwarizmi's: algorithms.

Addition: $O(n)$

Multiplication:

$$
\begin{array}{ccccccccccc}
 & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
 & \times & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
 & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
9 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 &
\end{array}
$$

More Al Khwarizmi's: algorithms.

Addition: $O(n)$

Multiplication:

$$
\begin{array}{ccccccccccc}
 & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
 & \times & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
 & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
9 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & \\
. & . & . & . & . & . & . & . & . & . & .
\end{array}
$$

More Al Khwarizmi's: algorithms.

Addition: $O(n)$

Multiplication:

$$
\begin{array}{cccccccccccc}
 & & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
 & & \times & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
 & & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
 & 9 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & \\
 & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & .
\end{array}
$$

More Al Khwarizmi's: algorithms.

Addition: $O(n)$

Multiplication:

$$
\begin{array}{ccccccccccccc}
 & & & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
 & & & \times & 9 & 2 & 1 & 2 & 3 & 7 & 6 & 9 & 1 \\
\hline
 & & & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
 & & 9 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & \\
 & & . & . & . & . & . & . & . & . & . & . & . \\
 & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & .
\end{array}
\quad n
$$

More Al Khwarizmi's: algorithms.

Addition: $O(n)$

Multiplication:

				1	2	3	4	5	6	7	8	9
			$\times$	9	2	1	2	3	7	6	9	1
				1	2	3	4	5	6	7	8	9
		9	2	2	2	2	2	2	2	2	1	
		.	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.

n

Time: $O(n^2)$

Multiplication

Multiplication: $O(n^2)$.

Multiplication

Multiplication: $O(n^2)$.

Is the best possible?

Multiplication

Multiplication: $O(n^2)$.

Is the best possible?

Every digit in x must multiply every digit in y at least once!

Multiplication

Multiplication: $O(n^2)$.

Is the best possible?

Every digit in x must multiply every digit in y at least once!
  $\Theta(n^2)$ such pairs.

Multiplication

Multiplication: $O(n^2)$.

Is the best possible?

Every digit in x must multiply every digit in y at least once!
  $\Theta(n^2)$ such pairs.

Is this the best possible?

Multiplication

Multiplication: $O(n^2)$.

Is the best possible?

Every digit in x must multiply every digit in y at least once!
 $\Theta(n^2)$ such pairs.

Is this the best possible?

(a) Yes.

(b) No.

Multiplication

Multiplication: $O(n^2)$.

Is the best possible?

Every digit in x must multiply every digit in y at least once!
  $\Theta(n^2)$ such pairs.

Is this the best possible?

(a) Yes.
(b) No.

No.

Multiplication

Multiplication: $O(n^2)$.

Is the best possible?

Every digit in x must multiply every digit in y at least once!
$\Theta(n^2)$ such pairs.

Is this the best possible?

(a) Yes.
(b) No.

No.
What ?!?!

Multiplication

Multiplication: $O(n^2)$.

Is the best possible?

Every digit in x must multiply every digit in y at least once!
  $\Theta(n^2)$ such pairs.

Is this the best possible?

(a) Yes.

(b) No.

No.
What ?!?!
Really!

Multiplication

Multiplication: $O(n^2)$.

Is the best possible?

Every digit in x must multiply every digit in y at least once!
$\Theta(n^2)$ such pairs.

Is this the best possible?

(a) Yes.
(b) No.

No.
What ?!?!
Really!
Later.

Exponentiation.

Compute x^y?

Exponentiation.

Compute x^y?

If x and y are n-bit numbers,
tightest valid upper bound on the number of bits for x^y...

Exponentiation.

Compute x^y?

If x and y are n-bit numbers,
tightest valid upper bound on the number of bits for x^y...

Exponentiation.

Compute x^y?

If x and y are n-bit numbers,
tightest valid upper bound on the number of bits for x^y...

(a) $2n$

Exponentiation.

Compute x^y?

If x and y are n-bit numbers,
tightest valid upper bound on the number of bits for x^y...

(a) $2n$

(b) n^2

Exponentiation.

Compute x^y?

If x and y are n-bit numbers,
tightest valid upper bound on the number of bits for x^y...

(a) $2n$

(b) n^2

(c) 2^n

(d) $n2^n$

Exponentiation.

Compute x^y?

If x and y are n-bit numbers,
tightest valid upper bound on the number of bits for x^y...

(a) $2n$

(b) n^2

(c) 2^n

(d) $n2^n$

Number of bits is $\log_2 x^y$

Exponentiation.

Compute x^y?

If x and y are n-bit numbers,
tightest valid upper bound on the number of bits for x^y...

(a) $2n$

(b) n^2

(c) 2^n

(d) $n2^n$

Number of bits is $\log_2 x^y = y \log x$

Exponentiation.

Compute x^y?

If x and y are n-bit numbers,
tightest valid upper bound on the number of bits for x^y...

(a) $2n$

(b) n^2

(c) 2^n

(d) $n2^n$

Number of bits is $\log_2 x^y = y \log x \leq 2^n \times n$.

Exponentiation.

Compute x^y?

If x and y are n-bit numbers,
tightest valid upper bound on the number of bits for x^y...

(a) $2n$

(b) n^2

(c) 2^n

(d) $n2^n$

Number of bits is $\log_2 x^y = y \log x \leq 2^n \times n$.

Does it make sense to do this?

Exponentiation.

Compute x^y?

If x and y are n-bit numbers,
tightest valid upper bound on the number of bits for x^y...

(a) $2n$

(b) n^2

(c) 2^n

(d) $n2^n$

Number of bits is $\log_2 x^y = y \log x \leq 2^n \times n$.

Does it make sense to do this?

Seems better just to keep x and y around.

Modular exponentiation.

Compute $x^y \pmod{z}$.

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c.

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.
Divide c by z, and output remainder.

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.
Divide c by z, and output remainder. $O(n^2)$.

Modular exponentiation.

Compute $x^y \pmod{z}$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.
Divide c by z, and output remainder. $O(n^2)$.

Addition: $(a+b) \pmod{z}$

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.
Divide c by z, and output remainder. $O(n^2)$.

Addition: $(a+b) \pmod z$ - add a, b maybe subtract z.

Modular exponentiation.

Compute $x^y \pmod{z}$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.
Divide c by z, and output remainder. $O(n^2)$.

Addition: $(a+b) \pmod{z}$ - add a, b maybe subtract z. $O(n)$

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.
Divide c by z, and output remainder. $O(n^2)$.

Addition: $(a+b) \pmod z$ - add a, b maybe subtract z. $O(n)$

Compute x^y, with $y-1$ modular multiplications with $O(n^2)$

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.
Divide c by z, and output remainder. $O(n^2)$.

Addition: $(a+b) \pmod z$ - add a, b maybe subtract z. $O(n)$

Compute x^y, with $y-1$ modular multiplications with $O(n^2)$

Still exponential!

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.
Divide c by z, and output remainder. $O(n^2)$.

Addition: $(a+b) \pmod z$ - add a, b maybe subtract z. $O(n)$

Compute x^y, with $y-1$ modular multiplications with $O(n^2)$

Still exponential! y is n bits so

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.
Divide c by z, and output remainder. $O(n^2)$.

Addition: $(a+b) \pmod z$ - add a, b maybe subtract z. $O(n)$

Compute x^y, with $y-1$ modular multiplications with $O(n^2)$

Still exponential! y is n bits so $y-1 = \Theta(2^n)$!

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.
Divide c by z, and output remainder. $O(n^2)$.

Addition: $(a+b) \pmod z$ - add a, b maybe subtract z. $O(n)$

Compute x^y, with $y-1$ modular multiplications with $O(n^2)$

Still exponential! y is n bits so $y-1 = \Theta(2^n)$!

Total is $O(n^2 2^n)$ time.

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.
Divide c by z, and output remainder. $O(n^2)$.

Addition: $(a+b) \pmod z$ - add a, b maybe subtract z. $O(n)$

Compute x^y, with $y-1$ modular multiplications with $O(n^2)$

Still exponential! y is n bits so $y-1 = \Theta(2^n)$!

Total is $O(n^2 2^n)$ time. Output is n-bits.

Modular exponentiation.

Compute $x^y \pmod z$.

x, y, z are n-bit numbers.

Then the result is an n-bit number!

Recall: Modular multiplication $a \cdot b \mod z$.

Multiply a and b to get c. $O(n^2)$.
Divide c by z, and output remainder. $O(n^2)$.

Addition: $(a+b) \pmod z$ - add a, b maybe subtract z. $O(n)$

Compute x^y, with $y-1$ modular multiplications with $O(n^2)$

Still exponential! y is n bits so $y - 1 = \Theta(2^n)$!

Total is $O(n^2 2^n)$ time. Output is n-bits.

Can we do better?

Repeated squaring

Compute $213^{87} \pmod{900}$?

Repeated squaring

Compute $213^{87} \pmod{900}$?

Notice:
$213^{87} = 213^{1+2+4+16+64} \pmod{900}$.

Repeated squaring

Compute $213^{87} \pmod{900}$?

Notice:
$213^{87} = 213^{1+2+4+16+64} \pmod{900}$.

$213^{87} = 213^1 \times 213^2 \times 213^4 \times 213^{16} \times 213^{64} \pmod{900}$.

Repeated squaring

Compute $213^{87} \pmod{900}$?

Notice:
$213^{87} = 213^{1+2+4+16+64} \pmod{900}$.

$213^{87} = 213^{1} \times 213^{2} \times 213^{4} \times 213^{16} \times 213^{64} \pmod{900}$.

87 in binary?

Repeated squaring

Compute $213^{87} \pmod{900}$?

Notice:
$213^{87} = 213^{1+2+4+16+64} \pmod{900}$.

$213^{87} = 213^1 \times 213^2 \times 213^4 \times 213^{16} \times 213^{64} \pmod{900}$.

87 in binary?
$87 \equiv 1010111$ in binary. 7 bits.

Repeated squaring

Compute $213^{87} \pmod{900}$?

Notice:
$213^{87} = 213^{1+2+4+16+64} \pmod{900}$.

$213^{87} = 213^1 \times 213^2 \times 213^4 \times 213^{16} \times 213^{64} \pmod{900}$.

87 in binary?
$87 \equiv 1010111$ in binary. 7 bits.

$213 \pmod{900}$
$213 \cdot 213 = 213^2 \pmod{900}$
$213^2 \cdot 213^2 = 213^4 \pmod{900}$

Repeated squaring

Compute $213^{87} \pmod{900}$?

Notice:
$213^{87} = 213^{1+2+4+16+64} \pmod{900}$.

$213^{87} = 213^1 \times 213^2 \times 213^4 \times 213^{16} \times 213^{64} \pmod{900}$.

87 in binary?
$87 \equiv 1010111$ in binary. 7 bits.

$213 \pmod{900}$
$213 \cdot 213 = 213^2 \pmod{900}$
$213^2 \cdot 213^2 = 213^4 \pmod{900}$

$\vdots$

Repeated squaring

Compute $213^{87} \pmod{900}$?

Notice:
$213^{87} = 213^{1+2+4+16+64} \pmod{900}$.

$213^{87} = 213^{1} \times 213^{2} \times 213^{4} \times 213^{16} \times 213^{64} \pmod{900}$.

87 in binary?
$87 \equiv 1010111$ in binary. 7 bits.

$213 \pmod{900}$
$213 \cdot 213 = 213^{2} \pmod{900}$
$213^{2} \cdot 213^{2} = 213^{4} \pmod{900}$

$\vdots$

$213^{32} \cdot 213^{32} = 213^{64} \mod 900$

Repeated squaring

Compute $213^{87} \pmod{900}$?

Notice:
$213^{87} = 213^{1+2+4+16+64} \pmod{900}$.

$213^{87} = 213^1 \times 213^2 \times 213^4 \times 213^{16} \times 213^{64} \pmod{900}$.

87 in binary?
$87 \equiv 1010111$ in binary. 7 bits.

$213 \pmod{900}$
$213 \cdot 213 = 213^2 \pmod{900}$
$213^2 \cdot 213^2 = 213^4 \pmod{900}$
$\vdots$
$213^{32} \cdot 213^{32} = 213^{64} \mod 900$

Only 6 (< 7) modular multiplications to compute the powers.

Repeated squaring

Compute $213^{87} \pmod{900}$?

Notice:
$213^{87} = 213^{1+2+4+16+64} \pmod{900}$.

$213^{87} = 213^1 \times 213^2 \times 213^4 \times 213^{16} \times 213^{64} \pmod{900}$.

87 in binary?
$87 \equiv 1010111$ in binary. 7 bits.

$213 \pmod{900}$
$213 \cdot 213 = 213^2 \pmod{900}$
$213^2 \cdot 213^2 = 213^4 \pmod{900}$
$\vdots$
$213^{32} \cdot 213^{32} = 213^{64} \mod 900$

Only 6 (< 7) modular multiplications to compute the powers.
At most 6 more modular multiplications to compute the result.

Repeated squaring

Compute $213^{87} \pmod{900}$?

Notice:
$213^{87} = 213^{1+2+4+16+64} \pmod{900}$.

$213^{87} = 213^{1} \times 213^{2} \times 213^{4} \times 213^{16} \times 213^{64} \pmod{900}$.

87 in binary?
$87 \equiv 1010111$ in binary. 7 bits.

$213 \pmod{900}$
$213 \cdot 213 = 213^2 \pmod{900}$
$213^2 \cdot 213^2 = 213^4 \pmod{900}$
$\vdots$
$213^{32} \cdot 213^{32} = 213^{64} \mod 900$

Only 6 (< 7) modular multiplications to compute the powers.
At most 6 more modular multiplications to compute the result.

Repeated squaring

Compute $213^{87} \pmod{900}$?

Notice:
$213^{87} = 213^{1+2+4+16+64} \pmod{900}$.

$213^{87} = 213^{1} \times 213^{2} \times 213^{4} \times 213^{16} \times 213^{64} \pmod{900}$.

87 in binary?
$87 \equiv 1010111$ in binary. 7 bits.

$213 \pmod{900}$
$213 \cdot 213 = 213^2 \pmod{900}$
$213^2 \cdot 213^2 = 213^4 \pmod{900}$
$\vdots$
$213^{32} \cdot 213^{32} = 213^{64} \mod 900$

Only 6 (< 7) modular multiplications to compute the powers.
At most 6 more modular multiplications to compute the result.

O(Number of bits) modular multiplications.

Repeated squaring.

General: x, y, z. (y is nonzero)

Repeated squaring.

General: x, y, z. (y is nonzero)

Compute $x^y \pmod{z}$.

Repeated squaring.

General: x, y, z. (y is nonzero)

Compute $x^y \pmod{z}$.

Write y in binary.

Repeated squaring.

General: x, y, z. (y is nonzero)

Compute $x^y \pmod{z}$.

Write y in binary.

Compute $x^1, x^2, x^4, \cdots, x^{2^n} \pmod{z}$ using n modular $\times$'s.

Repeated squaring.

General: x, y, z. (y is nonzero)

Compute $x^y \pmod{z}$.

Write y in binary.

Compute $x^1, x^2, x^4, \cdots, x^{2^n} \pmod{z}$ using n modular $\times$'s.

Multiply (at most) n products according to "set" bit positions in y.

Repeated squaring.

General: x, y, z. (y is nonzero)

Compute $x^y \pmod{z}$.

Write y in binary.

Compute $x^1, x^2, x^4, \cdots, x^{2^n} \pmod{z}$ using n modular $\times$'s.

Multiply (at most) n products according to "set" bit positions in y.

If y is n-bit number,

Repeated squaring.

General: x, y, z. (y is nonzero)

Compute $x^y \pmod{z}$.

Write y in binary.

Compute $x^1, x^2, x^4, \cdots, x^{2^n} \pmod{z}$ using n modular $\times$'s.

Multiply (at most) n products according to "set" bit positions in y.

If y is n-bit number,
.. $O(n)$ multiplications of n bit numbers each in time $O(n^2)$.

Repeated squaring.

General: x, y, z. (y is nonzero)

Compute $x^y \pmod{z}$.

Write y in binary.

Compute $x^1, x^2, x^4, \cdots, x^{2^n} \pmod{z}$ using n modular $\times$'s.

Multiply (at most) n products according to "set" bit positions in y.

If y is n-bit number,
.. $O(n)$ multiplications of n bit numbers each in time $O(n^2)$.

Total time is $O(n^3)$.

Repeated squaring.

General: x, y, z. (y is nonzero)

Compute $x^y \pmod{z}$.

Write y in binary.

Compute $x^1, x^2, x^4, \cdots, x^{2^n} \pmod{z}$ using n modular $\times$'s.

Multiply (at most) n products according to "set" bit positions in y.

If y is n-bit number,
.. $O(n)$ multiplications of n bit numbers each in time $O(n^2)$.

Total time is $O(n^3)$.

Coding it up.

```
def exp(x,y,z):
 if (y==1):
```

Coding it up.

```
def exp(x,y,z):
 if (y==1):
   return x
```

Coding it up.

```
def exp(x,y,z):
 if (y==1):
   return x
 else:
```

Coding it up.

$r = x^{y/2} \pmod{z}$

```
def exp(x,y,z):
 if (y==1):
   return x
 else:
   r = exp(x,y/2,z)
```

Coding it up.

$r = x^{y/2} \pmod z$

$v = (x^{y/2})^2 = x^{(\lfloor y/2 \rfloor \cdot 2} \pmod z$

```
def exp(x,y,z):
 if (y==1):
   return x
 else:
   r = exp(x,y/2,z)
   v = (r*r) %z
```

Coding it up.

$r = x^{y/2} \pmod z$

$v = (x^{y/2})^2 = x^{(\lfloor y/2 \rfloor \cdot 2} \pmod z$

Deal with $\lfloor y/2 \rfloor \cdot 2 \neq y$!

```
def exp(x,y,z):
 if (y==1):
   return x
 else:
   r = exp(x,y/2,z)
   v = (r*r) %z
```

Coding it up.

$r = x^{y/2} \pmod z$

$v = (x^{y/2})^2 = x^{(\lfloor y/2 \rfloor \cdot 2} \pmod z$

Deal with $\lfloor y/2 \rfloor \cdot 2 \neq y$!

odd y, $y = 1 + \lfloor y/2 \rfloor \cdot 2$.

$x^y = x^{1+\lfloor y/2 \rfloor \cdot 2} = x \cdot x^{\lfloor y/2 \rfloor \cdot 2} = x \cdot v$

```
def exp(x,y,z):
 if (y==1):
   return x
 else:
   r = exp(x,y/2,z)
   v = (r*r) %z
   if (odd(y)):
```

Coding it up.

$r = x^{y/2} \pmod z$

$v = (x^{y/2})^2 = x^{(\lfloor y/2 \rfloor \cdot 2} \pmod z$

Deal with $\lfloor y/2 \rfloor \cdot 2 \neq y$!

odd y, $y = 1 + \lfloor y/2 \rfloor \cdot 2$.

$x^y = x^{1+\lfloor y/2 \rfloor \cdot 2} = x \cdot x^{\lfloor y/2 \rfloor \cdot 2} = x \cdot v$

```
def exp(x,y,z):
 if (y==1):
   return x
 else:
   r = exp(x,y/2,z)
   v = (r*r) %z
   if (odd(y)):
     return (x * v) %z
   else:
```

Coding it up.

$r = x^{y/2} \pmod z$

$v = (x^{y/2})^2 = x^{(\lfloor y/2 \rfloor \cdot 2} \pmod z$

Deal with $\lfloor y/2 \rfloor \cdot 2 \neq y$!

odd y, $y = 1 + \lfloor y/2 \rfloor \cdot 2$.

$x^y = x^{1+\lfloor y/2 \rfloor \cdot 2} = x \cdot x^{\lfloor y/2 \rfloor \cdot 2} = x \cdot v$

```
def exp(x,y,z):
 if (y==1):
   return x
 else:
   r = exp(x,y/2,z)
   v = (r*r) %z
   if (odd(y)):
     return (x * v) %z
   else:
     return v
```

Modular Division.

What is division?

Modular Division.

What is division?

How do I divide by 5?

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$?

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x?

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x? 9×5

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x? $9 \times 5 = 45$,

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x? $9 \times 5 = 45$, $45 = 1 \pmod{11}$.

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x? $9 \times 5 = 45$, $45 = 1 \pmod{11}$. $x = 9$.

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x? $9 \times 5 = 45$, $45 = 1 \pmod{11}$. $x = 9$.

What is y?

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x? $9 \times 5 = 45$, $45 = 1 \pmod{11}$. $x = 9$.

What is y? $\quad 9(5y) = 9(7)$

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x? $9 \times 5 = 45$, $45 = 1 \pmod{11}$. $x = 9$.

What is y? $\quad 9(5y) = 9(7) = 63$

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x? $9 \times 5 = 45$, $45 = 1 \pmod{11}$. $x = 9$.

What is y? $\qquad 9(5y) = 9(7) = 63 = 8 \pmod{11}$.

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x? $9 \times 5 = 45$, $45 = 1 \pmod{11}$. $x = 9$.

What is y? $y = 9(5y) = 9(7) = 63 = 8 \pmod{11}$.

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x? $9 \times 5 = 45$, $45 = 1 \pmod{11}$. $x = 9$.

What is y? $y = 9(5y) = 9(7) = 63 = 8 \pmod{11}$.
Check: 5×8

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x? $9 \times 5 = 45$, $45 = 1 \pmod{11}$. $x = 9$.

What is y? $y = 9(5y) = 9(7) = 63 = 8 \pmod{11}$.
Check: $5 \times 8 = 40$

Modular Division.

What is division?

How do I divide by 5?

Multiply by $1/5$?

Multiplicative inverse of 5:

The number you multiply 5 by to get 1.
1 is the multiplicative identity.

Division $\equiv$ multiply by inverse.

Why divide?

$5y = 7 \pmod{11}$.

Find y?

Multiply both sides by multiplicative inverse of 5?

$5x = 1 \mod 11$? What is x? $9 \times 5 = 45$, $45 = 1 \pmod{11}$. $x = 9$.

What is y? $y = 9(5y) = 9(7) = 63 = 8 \pmod{11}$.
Check: $5 \times 8 = 40 = 7 \pmod{11}$.

Is there an inverse?

Inverse of $4 \pmod{6}$?

Is there an inverse?

Inverse of $4 \pmod{6}$?

No!

Is there an inverse?

Inverse of $4 \pmod 6$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

Is there an inverse?

Inverse of $4 \pmod 6$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

Is there an inverse?

Inverse of 4 $\pmod 6$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

gcd(x,y) - greatest common divisor of x and y.

Is there an inverse?

Inverse of $4 \pmod 6$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

gcd(x,y) - greatest common divisor of x and y.

gcd(x,y) $\neq$ 1 implies no inverse.

Is there an inverse?

Inverse of 4 $\pmod 6$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

gcd(x,y) - greatest common divisor of x and y.

gcd(x,y) $\neq 1$ implies no inverse.

Theorem:
$gcd(x,y) = d$, $d \geq 1 \rightarrow x$ has no multiplicative inverse modulo y.

Is there an inverse?

Inverse of 4 $\pmod 6$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

gcd(x,y) - greatest common divisor of x and y.

gcd(x,y) $\neq$ 1 implies no inverse.

Theorem:
$gcd(x, y) = d$, $d \geq 1 \rightarrow x$ has no multiplicative inverse modulo y.

Proof:

Is there an inverse?

Inverse of $4 \pmod 6$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

gcd(x,y) - greatest common divisor of x and y.

gcd(x,y) $\neq 1$ implies no inverse.

Theorem:
$gcd(x,y) = d$, $d \geq 1 \rightarrow x$ has no multiplicative inverse modulo y.

Proof: $ax = 1 \pmod y$ "$\equiv$"

Is there an inverse?

Inverse of $4 \pmod 6$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

gcd(x,y) - greatest common divisor of x and y.

gcd(x,y) $\neq 1$ implies no inverse.

Theorem:
$gcd(x, y) = d$, $d \geq 1 \rightarrow x$ has no multiplicative inverse modulo y.

Proof: $ax = 1 \pmod y$ "$\equiv$" $ax = 1 + by$ for integer b

Is there an inverse?

Inverse of $4 \pmod 6$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

gcd(x,y) - greatest common divisor of x and y.

gcd(x,y) $\neq 1$ implies no inverse.

Theorem:
$gcd(x, y) = d$, $d \geq 1 \rightarrow x$ has no multiplicative inverse modulo y.

Proof: $ax = 1 \pmod y$ "$\equiv$" $ax = 1 + by$ for integer b

$ax - by = 1$.

Is there an inverse?

Inverse of $4 \pmod 6$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

gcd(x,y) - greatest common divisor of x and y.

gcd(x,y) $\neq 1$ implies no inverse.

Theorem:
$gcd(x,y) = d$, $d \geq 1 \rightarrow x$ has no multiplicative inverse modulo y.

Proof: $ax = 1 \pmod y$ "$\equiv$" $ax = 1 + by$ for integer b

$ax - by = 1$. $x = id, y = jd$ since d divides both.

Is there an inverse?

Inverse of $4 \pmod 6$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

gcd(x,y) - greatest common divisor of x and y.

gcd(x,y) $\neq 1$ implies no inverse.

Theorem:
$gcd(x,y) = d$, $d \geq 1 \rightarrow x$ has no multiplicative inverse modulo y.

Proof: $ax = 1 \pmod y$ "$\equiv$" $ax = 1 + by$ for integer b

$ax - by = 1$. $x = id, y = jd$ since d divides both.

$a(id) - b(jd) = 1 \rightarrow d(ia - jb) = 1$.

Is there an inverse?

Inverse of 4 $\pmod{6}$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

gcd(x,y) - greatest common divisor of x and y.

gcd(x,y) $\neq 1$ implies no inverse.

Theorem:
$gcd(x,y) = d$, $d \geq 1 \rightarrow x$ has no multiplicative inverse modulo y.

Proof: $ax = 1 \pmod{y}$ "$\equiv$" $ax = 1 + by$ for integer b

$ax - by = 1$. $x = id, y = jd$ since d divides both.

$a(id) - b(jd) = 1 \rightarrow d(ia - jb) = 1$.

d must be a factor of 1.

Is there an inverse?

Inverse of 4 $\pmod 6$?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

gcd(x,y) - greatest common divisor of x and y.

gcd(x,y) $\neq 1$ implies no inverse.

Theorem:
$gcd(x,y) = d$, $d \geq 1 \rightarrow x$ has no multiplicative inverse modulo y.

Proof: $ax = 1 \pmod y$ "$\equiv$" $ax = 1 + by$ for integer b

$ax - by = 1$. $x = id, y = jd$ since d divides both.

$a(id) - b(jd) = 1 \rightarrow d(ia - jb) = 1$.

d must be a factor of 1. That is, $d = 1$.

Is there an inverse?

Inverse of 4 (mod 6)?

No!

$4j$ is at least 2 away from $6k$ for any j, k.

They have a common divisor that is greater than 1.

gcd(x,y) - greatest common divisor of x and y.

gcd(x,y) $\neq 1$ implies no inverse.

Theorem:
$gcd(x, y) = d$, $d \geq 1 \rightarrow x$ has no multiplicative inverse modulo y.

Proof: $ax = 1 \pmod{y}$ "$\equiv$" $ax = 1 + by$ for integer b

$ax - by = 1$. $x = id, y = jd$ since d divides both.

$a(id) - b(jd) = 1 \rightarrow d(ia - jb) = 1$.

d must be a factor of 1. That is, $d = 1$. $\square$

Review: extended euclid's algorithm and inverses.

Extended GCD:

Review: extended euclid's algorithm and inverses.

Extended GCD:

Given x, y.

Review: extended euclid's algorithm and inverses.

Extended GCD:

Given x, y.
Returns: (d, a, b) where $ax + by = d$, and $d = gcd(x, y)$

Review: extended euclid's algorithm and inverses.

Extended GCD:

Given x, y.
Returns: (d, a, b) where $ax + by = d$, and $d = gcd(x, y)$

Find inverse of x modulo N, if $gcd(x, N) = 1$?

Review: extended euclid's algorithm and inverses.

Extended GCD:

Given x, y.
Returns: (d, a, b) where $ax + by = d$, and $d = gcd(x, y)$

Find inverse of x modulo N, if $gcd(x, N) = 1$?

(A) Run Euclid on x, N, output a.
(B) Run Euclid on x, N, output b.

Review: extended euclid's algorithm and inverses.

Extended GCD:

Given x, y.
Returns: (d, a, b) where $ax + by = d$, and $d = gcd(x, y)$

Find inverse of x modulo N, if $gcd(x, N) = 1$?

(A) Run Euclid on x, N, output a.

(B) Run Euclid on x, N, output b.

A. $1 = ax + bN$

Review: extended euclid's algorithm and inverses.

Extended GCD:

Given x, y.
Returns: (d, a, b) where $ax + by = d$, and $d = gcd(x, y)$

Find inverse of x modulo N, if $gcd(x, N) = 1$?

(A) Run Euclid on x, N, output a.

(B) Run Euclid on x, N, output b.

A. $1 = ax + bN = ax \pmod{N}$,

Review: extended euclid's algorithm and inverses.

Extended GCD:

Given x, y.
Returns: (d, a, b) where $ax + by = d$, and $d = gcd(x, y)$

Find inverse of x modulo N, if $gcd(x, N) = 1$?

(A) Run Euclid on x, N, output a.

(B) Run Euclid on x, N, output b.

A. $1 = ax + bN = ax \pmod{N}$,
so a is multiplicative inverse of x modulo N.

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d, a, b); d = ax + by$.

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d, a, b); d = ax + by$.

$x =$ __________

$y =$ ____

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d,a,b); d = ax + by$.

$x =$ __________

$y =$ ____

Get "close" to y with x's:

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d, a, b); d = ax + by$.

$x =$__________

$y =$____

Get “close” to y with x’s:

$kx =$____

$y =$ ____

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d,a,b); d = ax + by$.

$x =$ __________

$y =$ ____

Get "close" to y with x's:

$kx =$ ____

$y =$ ____

$k = \lfloor y/x \rfloor$ (Use long division.) (Time: $O(n^2)$ time.)

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d,a,b); d = ax + by$.

$x =$__________

$y =$____

Get “close” to y with x’s:

$kx =$____

$y =$ ____

$k = \lfloor y/x \rfloor$ (Use long division.) (Time: $O(n^2)$ time.)

$(y - kx)$ preserves common divisor!

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d,a,b); d = ax+by$.

$x =$__________

$y =$____

Get “close” to y with x’s:

$kx =$____

$y =$ ____

$k = \lfloor y/x \rfloor$ (Use long division.) (Time: $O(n^2)$ time.)

$(y-kx)$ preserves common divisor!
Anything that divides both x and y, divides $(y-kx)$

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d,a,b); d = ax + by$.

$x =$ __________

$y =$ ____

Get "close" to y with x's:

$kx =$ ____

$y =$ ____

$k = \lfloor y/x \rfloor$ (Use long division.) (Time: $O(n^2)$ time.)

$(y - kx)$ preserves common divisor!
Anything that divides both x and y, divides $(y - kx)$

Recurse for $y - kx$ and x

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d,a,b); d = ax + by$.

$x =$_________

$y =$____

Get "close" to y with x's:

$kx =$____

$y =$ ____

$k = \lfloor y/x \rfloor$ (Use long division.) (Time: $O(n^2)$ time.)

$(y - kx)$ preserves common divisor!
Anything that divides both x and y, divides $(y - kx)$

Recurse for $y - kx$ and x
$d|x$ and $d|(y - kx)$

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d,a,b); d = ax + by$.

$x =$ __________

$y =$ ____

Get “close” to y with x’s:

$kx =$ ____

$y =$ ____

$k = \lfloor y/x \rfloor$ (Use long division.) (Time: $O(n^2)$ time.)

$(y - kx)$ preserves common divisor!
Anything that divides both x and y, divides $(y - kx)$

Recurse for $y - kx$ and x
$d|x$ and $d|(y - kx)$ Also $d'|x$ and $d'|(y - kx)$

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d, a, b); d = ax + by$.

$x =$_________

$y =$____

Get "close" to y with x's:

$kx =$____

$y =$ ____

$k = \lfloor y/x \rfloor$ (Use long division.) (Time: $O(n^2)$ time.)

$(y - kx)$ preserves common divisor!
Anything that divides both x and y, divides $(y - kx)$

Recurse for $y - kx$ and x
$d|x$ and $d|(y - kx)$ Also $d'|x$ and $d'|(y - kx) \implies d'|y$.

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d, a, b); d = ax + by$.

$x =$ __________

$y =$ ____

Get "close" to y with x's:

$kx =$ ____

$y =$ ____

$k = \lfloor y/x \rfloor$ (Use long division.) (Time: $O(n^2)$ time.)

$(y - kx)$ preserves common divisor!
Anything that divides both x and y, divides $(y - kx)$

Recurse for $y - kx$ and x
$d|x$ and $d|(y - kx)$ Also $d'|x$ and $d'|(y - kx) \implies d'|y$.
$\rightarrow gcd(x, y) = gcd(x, y - kx)$.

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d,a,b); d = ax + by$.

$x =$ __________

$y =$ ____

Get "close" to y with x's:

$kx =$ ____

$y =$ ____

$k = \lfloor y/x \rfloor$ (Use long division.) (Time: $O(n^2)$ time.)

$(y - kx)$ preserves common divisor!
Anything that divides both x and y, divides $(y - kx)$

Recurse for $y - kx$ and x
$d|x$ and $d|(y - kx)$ Also $d'|x$ and $d'|(y - kx) \implies d'|y$.
$\rightarrow gcd(x,y) = gcd(x, y - kx)$.

Get (d, a', b') where $d = a'(y - kx) + b'x$

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d,a,b); d = ax + by$.

$x =$ __________

$y =$ ____

Get “close” to y with x’s:

$kx =$ ____

$y =$ ____

$k = \lfloor y/x \rfloor$ (Use long division.) (Time: $O(n^2)$ time.)

$(y-kx)$ preserves common divisor!
Anything that divides both x and y, divides $(y-kx)$

Recurse for $y-kx$ and x
$d|x$ and $d|(y-kx)$ Also $d'|x$ and $d'|(y-kx) \implies d'|y$.
$\rightarrow gcd(x,y) = gcd(x,y-kx)$.

Get (d,a',b') where $d = a'(y-kx)+b'x = (b'-ka')x + a'y$.

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d, a, b); d = ax + by$.

$x =$__________

$y =$____

Get "close" to y with x's:

$kx =$____

$y =$ ____

$k = \lfloor y/x \rfloor$ (Use long division.) (Time: $O(n^2)$ time.)

$(y - kx)$ preserves common divisor!
Anything that divides both x and y, divides $(y - kx)$

Recurse for $y - kx$ and x
$d|x$ and $d|(y - kx)$ Also $d'|x$ and $d'|(y - kx) \implies d'|y$.
$\rightarrow gcd(x, y) = gcd(x, y - kx)$.

Get (d, a', b') where $d = a'(y - kx) + b'x = (b' - ka')x + a'y$.

Return $(d, b' - ka', a')$.

Extended Euclid/Correctness.

Ext-gcd(x,y): $(d, a, b); d = ax + by$.

$x =$__________

$y =$____

Get "close" to y with x's:

$kx =$____

$y =$ ____

$k = \lfloor y/x \rfloor$ (Use long division.) (Time: $O(n^2)$ time.)

$(y - kx)$ preserves common divisor!
Anything that divides both x and y, divides $(y - kx)$

Recurse for $y - kx$ and x
$d|x$ and $d|(y - kx)$ Also $d'|x$ and $d'|(y - kx) \implies d'|y$.
$\rightarrow gcd(x, y) = gcd(x, y - kx)$.

Get (d, a', b') where $d = a'(y - kx) + b'x = (b' - ka')x + a'y$.

Return $(d, b' - ka', a')$.
Time for one recursive call: $O(n^2)$.

See you Friday..

....