Warmup

Usually, we express numbers in base 10, meaning we represent it as integer combinations of powers of 10, with coefficients from $\{0, 1, \ldots, 9\}$:

$$327 = 3 \cdot 10^2 + 2 \cdot 10 + 7 \cdot 1.$$

To express a number in base 2, you write it as the integer combination of powers of 2, with coefficients from $\{0, 1\}$:

n	0	1	2	3	4	5	6	7	8	9	10	11
2^n	1	2	4	8	16	32	64	128	256	512	1024	2048

For example,

$$\begin{aligned} 327 &= 2^8 + 71 = 2^8 + 2^6 + 7 \\ &= 2^8 + 2^6 + 2^2 + 2 + 1 = 101000111 \text{ (base 2)}. \end{aligned}$$

You try: Write 10, 100, and 1000 in base 2.

Warmup

Usually, we express numbers in base 10, meaning we represent it as integer combinations of powers of 10, with coefficients from $\{0, 1, \ldots, 9\}$:

$$327 = 3 \cdot 10^2 + 2 \cdot 10 + 7 \cdot 1.$$

To express a number in base 2, you write it as the integer combination of powers of 2, with coefficients from $\{0, 1\}$:

n	0	1	2	3	4	5	6	7	8	9	10	11
2^n	1	2	4	8	16	32	64	128	256	512	1024	2048

For example,

$$\begin{aligned} 327 &= 2^8 + 71 = 2^8 + 2^6 + 7 \\ &= 2^8 + 2^6 + 2^2 + 2 + 1 = 101000111 \text{ (base 2)}. \end{aligned}$$

You try: Write 10, 100, and 1000 in base 2.

$$10 = 2^3 + 2 = 1010 \text{ (base 2)}$$

$$100 = 2^6 + 2^5 + 2^2 = 1100100 \text{ (base 2)}$$

$$1000 = 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3 = 1111101000 \text{ (base 2)}$$

Computational methods

Example: Compute $7^a \pmod{853}$.

Computational methods

Example: Compute $7^a \pmod{853}$.

Since 853 is prime, we have $\gcd(7, 853) = 1$. So

$$\phi(853) = 853 - 1, \quad \text{and} \quad 7^{852} \equiv 1 \pmod{853}.$$

Computational methods

Example: Compute $7^a \pmod{853}$.

Since 853 is prime, we have $\gcd(7, 853) = 1$. So

$$\phi(853) = 853 - 1, \quad \text{and} \quad 7^{852} \equiv 1 \pmod{853}.$$

But how *useful* is Fermat's little theorem here?

Computational methods

Example: Compute $7^a \pmod{853}$.

Since 853 is prime, we have $\gcd(7, 853) = 1$. So

$$\phi(853) = 853 - 1, \quad \text{and} \quad 7^{852} \equiv 1 \pmod{853}.$$

But how *useful* is Fermat's little theorem here?

Not very...

a	7^a	$7^a \pmod{853}$
1	7	7
2	49	49
3	343	343
4	2401	695
5	16807	600
6	117649	788
7	823543	398
8	5764801	227
9	40353607	736
10	282475249	34

a	7^a	$7^a \pmod{853}$
11	1977326743	238
12	13841287201	813
13	96889010407	573
14	$6.78223E+11$	599
15	$4.74756E+12$	781
16	$3.32329E+13$	349
17	$2.32631E+14$	737
18	$1.62841E+15$	$\#NUM!$
19	$1.13989E+16$	$\#NUM!$
20	$7.97923E+16$	$\#NUM!$

Computational methods

We can get some leverage out of successively simplifying: If

$$x^{a-1} \equiv_n r \quad \text{then} \quad x^a \equiv_n xr.$$

Computational methods

We can get some leverage out of successively simplifying: If

$$x^{a-1} \equiv_n r \quad \text{then} \quad x^a \equiv_n xr.$$

a	$7 \cdot \overline{7^{a-1}}$	$7^a \pmod{853}$	a	$7 \cdot \overline{7^{a-1}}$	$7^a \pmod{853}$
1	7	7	11	238	238
2	49	49	12	1666	813
3	343	343	13	5691	573
4	2401	695	14	4011	599
5	4865	600	15	4193	781
6	4200	788	16	5467	349
7	5516	398	17	2443	737
8	2786	227	18	5159	41
9	1589	736	19	287	287
10	5152	34	20	2009	303

Advantage: computations are possible

Disadvantage: must compute 7^b for $b = 1, 2, \ldots, a-1$ to get 7^a.

Computational methods

Next simplification: successive squaring. Namely, if

$$x^{2^{b-1}} \equiv_n r \quad \text{then} \quad x^{2^b} = (x^{2^{b-1}})^2 \equiv_n r^2.$$

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Computational methods

Next simplification: successive squaring. Namely, if

$$x^{2^{b-1}} \equiv_n r \quad \text{then} \quad x^{2^b} = (x^{2^{b-1}})^2 \equiv_n r^2.$$

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Ex: Use this data to compute $7^{327} \pmod{853}$:

Computational methods

Next simplification: successive squaring. Namely, if

$$x^{2^{b-1}} \equiv_n r \quad \text{then} \quad x^{2^b} = (x^{2^{b-1}})^2 \equiv_n r^2.$$

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Ex: Use this data to compute $7^{327} \pmod{853}$:

Recall

$$327 = 2^8 + 2^6 + 2^2 + 2 + 1.$$

Computational methods

Next simplification: successive squaring. Namely, if

$$x^{2^{b-1}} \equiv_n r \quad \text{then} \quad x^{2^b} = (x^{2^{b-1}})^2 \equiv_n r^2.$$

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Ex: Use this data to compute

$$7^{327} \pmod{853}:$$

Recall

$$327 = 2^8 + 2^6 + 2^2 + 2 + 1.$$

So

$$\begin{aligned} 7^{327} &= 7^{(2^8)}7^{(2^6)}7^{(2^2)}7^2 7 \\ &\equiv_{853} 298 \cdot 123 \cdot 695 \cdot 49 \cdot 7 \\ &= 8737763790 \end{aligned}$$

Computational methods

Next simplification: successive squaring. Namely, if

$$x^b \equiv_n r \quad \text{then} \quad x^{2b} = (x^b)^2 \equiv_n r^2.$$

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Ex: Use this data to compute $7^{327} \pmod{853}$:

Last simplification:

reduce successive products.

We had

$$7^{327} \equiv_{853} 298 \cdot 123 \cdot 695 \cdot 49 \cdot 7$$

Computational methods

Next simplification: successive squaring. Namely, if

$$x^b \equiv_n r \quad \text{then} \quad x^{2b} = (x^b)^2 \equiv_n r^2.$$

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Ex: Use this data to compute

$$7^{327} \pmod{853}:$$

Last simplification:

reduce successive products.

We had

$$7^{327} \equiv_{853} 298 \cdot 123 \cdot 695 \cdot 49 \cdot 7$$

Now,

$$298 \cdot 123 = 36654 \equiv_{853} 828$$

Computational methods

Next simplification: successive squaring. Namely, if

$$x^b \equiv_n r \quad \text{then} \quad x^{2b} = (x^b)^2 \equiv_n r^2.$$

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Ex: Use this data to compute

$$7^{327} \pmod{853}:$$

Last simplification:

reduce successive products.

We had

$$7^{327} \equiv_{853} 298 \cdot 123 \cdot 695 \cdot 49 \cdot 7$$

Now,

$$298 \cdot 123 = 36654 \equiv_{853} 828$$

$$828 \cdot 695 = 575460 \equiv_{853} 538$$

Computational methods

Next simplification: successive squaring. Namely, if

$$x^b \equiv_n r \quad \text{then} \quad x^{2b} = (x^b)^2 \equiv_n r^2.$$

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Ex: Use this data to compute $7^{327} \pmod{853}$:

Last simplification:

reduce successive products.

We had

$$7^{327} \equiv_{853} 298 \cdot 123 \cdot 695 \cdot 49 \cdot 7$$

Now,

$$298 \cdot 123 = 36654 \equiv_{853} 828$$

$$828 \cdot 695 = 575460 \equiv_{853} 538$$

$$538 \cdot 49 = 26362 \equiv_{853} 772$$

Computational methods

Next simplification: successive squaring. Namely, if

$$x^b \equiv_n r \quad \text{then} \quad x^{2b} = (x^b)^2 \equiv_n r^2.$$

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Ex: Use this data to compute

$$7^{327} \pmod{853}:$$

Last simplification:

reduce successive products.

We had

$$7^{327} \equiv_{853} 298 \cdot 123 \cdot 695 \cdot 49 \cdot 7$$

Now,

$$298 \cdot 123 = 36654 \equiv_{853} 828$$

$$828 \cdot 695 = 575460 \equiv_{853} 538$$

$$538 \cdot 49 = 26362 \equiv_{853} 772$$

$$772 \cdot 7 = 26362 \equiv_{853} 286$$

Computational methods

Next simplification: successive squaring. Namely, if

$$x^b \equiv_n r \quad \text{then} \quad x^{2b} = (x^b)^2 \equiv_n r^2.$$

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Ex: Use this data to compute $7^{327} \pmod{853}$:

Last simplification:

reduce successive products.

We had

$$7^{327} \equiv_{853} 298 \cdot 123 \cdot 695 \cdot 49 \cdot 7$$

Now,

$$298 \cdot 123 = 36654 \equiv_{853} 828$$

$$828 \cdot 695 = 575460 \equiv_{853} 538$$

$$538 \cdot 49 = 26362 \equiv_{853} 772$$

$$772 \cdot 7 = 26362 \equiv_{853} 286$$

So $7^{327} \equiv_{853} \boxed{286}$

Computational methods

Next simplification: successive squaring. Namely, if

$$x^b \equiv_n r \quad \text{then} \quad x^{2b} = (x^b)^2 \equiv_n r^2.$$

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Ex: Use this data to compute $7^{327} \pmod{853}$:

Last simplification:

reduce successive products.

We had

$$7^{327} \equiv_{853} 298 \cdot 123 \cdot 695 \cdot 49 \cdot 7$$

Now,

$$298 \cdot 123 = 36654 \equiv_{853} 828$$

$$828 \cdot 695 = 575460 \equiv_{853} 538$$

$$538 \cdot 49 = 26362 \equiv_{853} 772$$

$$772 \cdot 7 = 26362 \equiv_{853} 286$$

So $7^{327} \equiv_{853} \boxed{286}$

You try: Compute $7^{100} \pmod{853}$ using only the data above and basic calculator functions.

Computational methods

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

How many data points do we *need*?

Computational methods

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

How many data points do we *need*?

Back to Fermat/Euler:

$$7^{852} \equiv 1 \pmod{853}$$

Computational methods

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

How many data points do we *need*?

Back to Fermat/Euler:

$$7^{852} \equiv 1 \pmod{853}$$

So we don't need powers of 7 larger than 7^{851}. So, since

$$2^9 = 512 < 851$$

and

$$2^{10} = 1024 > 851,$$

the data to the left is *exactly* what we need.

Computational methods

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Example: Compute $7^{1000} \pmod{853}$.

How many data points do we *need*?

Back to Fermat/Euler:

$$7^{852} \equiv 1 \pmod{853}$$

So we don't need powers of 7 larger than 7^{851}. So, since

$$2^9 = 512 < 851$$

and

$$2^{10} = 1024 > 851,$$

the data to the left is *exactly* what we need.

Computational methods

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Example: Compute $7^{1000} \pmod{853}$.

We have $1000 \equiv_{852} 148$

How many data points do we *need*?

Back to Fermat/Euler:

$$7^{852} \equiv 1 \pmod{853}$$

So we don't need powers of 7 larger than 7^{851}. So, since

$$2^9 = 512 < 851$$

and

$$2^{10} = 1024 > 851,$$

the data to the left is *exactly* what we need.

Computational methods

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

Example: Compute $7^{1000} \pmod{853}$.

We have $1000 \equiv_{852} 148$, so $7^{1000} \equiv_{853} 7^{148}$.

How many data points do we *need*?

Back to Fermat/Euler:

$$7^{852} \equiv 1 \pmod{853}$$

So we don't need powers of 7 larger than 7^{851}. So, since

$$2^9 = 512 < 851$$

and

$$2^{10} = 1024 > 851,$$

the data to the left is *exactly* what we need.

Computational methods

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

How many data points do we *need*?

Back to Fermat/Euler:

$$7^{852} \equiv 1 \pmod{853}$$

So we don't need powers of 7 larger than 7^{851}. So, since

$$2^9 = 512 < 851$$

and

$$2^{10} = 1024 > 851,$$

the data to the left is *exactly* what we need.

Example: Compute $7^{1000} \pmod{853}$.

We have $1000 \equiv_{852} 148$, so $7^{1000} \equiv_{853} 7^{148}$. Next,

$$148 = 128 + 16 + 4 = 2^7 + 2^4 + 2^2.$$

Computational methods

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

How many data points do we *need*?

Back to Fermat/Euler:

$$7^{852} \equiv 1 \pmod{853}$$

So we don't need powers of 7 larger than 7^{851}. So, since

$$2^9 = 512 < 851$$

and

$$2^{10} = 1024 > 851,$$

the data to the left is *exactly* what we need.

Example: Compute $7^{1000} \pmod{853}$.

We have $1000 \equiv_{852} 148$, so $7^{1000} \equiv_{853} 7^{148}$. Next,

$$148 = 128 + 16 + 4 = 2^7 + 2^4 + 2^2.$$

So

$$7^{1000} \equiv_{853} 7^{148} = 7^{(2^7)} 7^{(2^4)} 7^{(2^2)}$$

Computational methods

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

How many data points do we *need*?

Back to Fermat/Euler:

$$7^{852} \equiv 1 \pmod{853}$$

So we don't need powers of 7 larger than 7^{851}. So, since

$$2^9 = 512 < 851$$

and

$$2^{10} = 1024 > 851,$$

the data to the left is *exactly* what we need.

Example: Compute $7^{1000} \pmod{853}$.

We have $1000 \equiv_{852} 148$, so $7^{1000} \equiv_{853} 7^{148}$. Next,

$$148 = 128 + 16 + 4 = 2^7 + 2^4 + 2^2.$$

So

$$7^{1000} \equiv_{853} 7^{148} = 7^{(2^7)}7^{(2^4)}7^{(2^2)} \equiv_{853} 628 \cdot 349 \cdot 695$$

Computational methods

2^a	$\overline{7^{2^{a-1}}}^2$	$7^{2^a} \pmod{853}$
1	7	7
2	49	49
2^2	2401	695
2^3	483025	227
2^4	51529	349
2^5	121801	675
2^6	455625	123
2^7	15129	628
2^8	394384	298
2^9	88804	92

How many data points do we *need*?

Back to Fermat/Euler:

$$7^{852} \equiv 1 \pmod{853}$$

So we don't need powers of 7 larger than 7^{851}. So, since

$$2^9 = 512 < 851$$

and

$$2^{10} = 1024 > 851,$$

the data to the left is *exactly* what we need.

Example: Compute $7^{1000} \pmod{853}$.

We have $1000 \equiv_{852} 148$, so $7^{1000} \equiv_{853} 7^{148}$. Next,

$$148 = 128 + 16 + 4 = 2^7 + 2^4 + 2^2.$$

So

$$7^{1000} \equiv_{853} 7^{148} = 7^{(2^7)}7^{(2^4)}7^{(2^2)} \equiv_{853} 628 \cdot 349 \cdot 695 \equiv_{853} 804 \cdot 695 \equiv_{853} \boxed{65}$$

Application: prime testing

Question: How can we tell if a (very large) number m is prime or not?

Application: prime testing

Question: How can we tell if a (very large) number m is prime or not?

Definitive answer: Divide m by every number between 2 and $\sqrt{m}$, and check.

Application: prime testing

Question: How can we tell if a (very large) number m is prime or not?

Definitive answer: Divide m by every number between 2 and $\sqrt{m}$, and check. Slow!!

Test for not primes: If m is prime, then $a^{m-1} \equiv_m 1$ for any a relatively prime to m.

Application: prime testing

Question: How can we tell if a (very large) number m is prime or not?

Definitive answer: Divide m by every number between 2 and $\sqrt{m}$, and check. Slow!!

Test for not primes: If m is prime, then $a^{m-1} \equiv_m 1$ for any a relatively prime to m. So pick an $a < m$, and compute $\gcd(a, m)$.

Application: prime testing

Question: How can we tell if a (very large) number m is prime or not?

Definitive answer: Divide m by every number between 2 and $\sqrt{m}$, and check. Slow!!

Test for not primes: If m is prime, then $a^{m-1} \equiv_m 1$ for any a relatively prime to m. So pick an $a < m$, and compute $\gcd(a, m)$.

1. If $\gcd(a, m) > 1$, then m is not prime, and we're done.

Application: prime testing

Question: How can we tell if a (very large) number m is prime or not?

Definitive answer: Divide m by every number between 2 and $\sqrt{m}$, and check. Slow!!

Test for not primes: If m is prime, then $a^{m-1} \equiv_m 1$ for any a relatively prime to m. So pick an $a < m$, and compute $\gcd(a, m)$.

1. If $\gcd(a, m) > 1$, then m is not prime, and we're done.
2. Otherwise, if $\gcd(a, m) = 1$, compute $a^{m-1} \pmod{m}$ using successive squaring method.

Application: prime testing

Question: How can we tell if a (very large) number m is prime or not?

Definitive answer: Divide m by every number between 2 and $\sqrt{m}$, and check. Slow!!

Test for not primes: If m is prime, then $a^{m-1} \equiv_m 1$ for any a relatively prime to m. So pick an $a < m$, and compute $\gcd(a, m)$.

1. If $\gcd(a, m) > 1$, then m is not prime, and we're done.
2. Otherwise, if $\gcd(a, m) = 1$, compute $a^{m-1} \pmod{m}$ using successive squaring method.
   (i) If $a^{m-1} \not\equiv_m 1$, then m is not prime, and we're done.

Application: prime testing

Question: How can we tell if a (very large) number m is prime or not?

Definitive answer: Divide m by every number between 2 and $\sqrt{m}$, and check. Slow!!

Test for not primes: If m is prime, then $a^{m-1} \equiv_m 1$ for any a relatively prime to m. So pick an $a < m$, and compute $\gcd(a, m)$.

1. If $\gcd(a, m) > 1$, then m is not prime, and we're done.
2. Otherwise, if $\gcd(a, m) = 1$, compute $a^{m-1} \pmod{m}$ using successive squaring method.
   - (i) If $a^{m-1} \not\equiv_m 1$, then m is not prime, and we're done.
   - (ii) If $a^{m-1} \equiv_m 1$, then m may or may not be prime, and the test is inconclusive. (See exercise 23(c))