How a computer can prove a number is composite or prime without factoring it. The n’s below appear in Exercise 19.8 of our book.

\[ n = 285707540662569884530199015485750433489 \]

Since \(2^{(n-1)}\) is not congruent to 1 (mod n) we immediately have that n is COMPOSITE

\[ n = 285707540662569884530199015485751094149 \]

n is starting to look like a prime, but since it could be a Carmichael number we switch the Rabin-Miller test to try to see if we can show it is composite.
\[ n - 1 = 2^2 \cdot q \] with

\begin{align*}
\text{In}[60]:= & \quad q = 71426885165642471132549753871437773537 \\
\text{Out}[60]= & \quad 71426885165642471132549753871437773537
\end{align*}

We try the Rabin-Miller Test for \( a = 2, 3, 5, 7 \)

\begin{align*}
\text{In}[61]:= & \quad \text{PowerMod}[2, 2 \cdot q, n] \\
\text{Out}[61]= & \quad 285707540662569884530199015485751094148 \\
\text{In}[62]:= & \quad n - 1 \\
\text{Out}[62]= & \quad 285707540662569884530199015485751094148
\end{align*}

This is just \(-1 (\text{mod } n)\). We can’t go further with \( a = 2 \). We try \( a = 3 \).

\begin{align*}
\text{In}[63]:= & \quad \text{PowerMod}[3, 2 \cdot q, n] \\
\text{Out}[63]= & \quad 285707540662569884530199015485751094148
\end{align*}

This is \(-1 (\text{mod } n)\) again. We can’t go further with \( a = 3 \). We try \( a = 5 \).

\begin{align*}
\text{In}[64]:= & \quad \text{PowerMod}[5, 2 \cdot q, n] \\
\text{Out}[64]= & \quad 1
\end{align*}

\begin{align*}
\text{In}[65]:= & \quad \text{PowerMod}[5, q, n] \\
\text{Out}[65]= & \quad 285707540662569884530199015485751094148
\end{align*}

This is \(-1 (\text{mod } n)\) again. So \( a = 5 \) doesn’t work either. We try \( a = 7 \).

\begin{align*}
\text{In}[66]:= & \quad \text{PowerMod}[7, 2 \cdot q, n] \\
\text{Out}[66]= & \quad 1
\end{align*}
We get \(-1 \pmod{n}\) again. Since the Rabin–Miller test failed for \(a = 2, 3, 5, 7\) we begin to think that \(n\) could be prime. We try to show it is prime. We factor \(n - 1\) (Note \(n - 1\) is even, so it has a power of 2 as a factor, but we are left with factoring \(q\) odd and large. If \(q\) has some small factors this method works well).

\[
\text{In[68]:=} \quad \text{FactorInteger}[n - 1]
\]
\[
\text{Out[68]=} \quad \{\{2, 2\}, \{1476241557300827, 1\}, \{48384280209696856047731, 1\}\}
\]

\(n - 1\) is the product of three primes, 2, \(p_1 = 1476241557300827\), and \(p_2 = 48384280209696856047731\).

We apply the primitive root test to see if we can find a primitive root.

\[
\text{In[70]:=} \quad p_1 = 1476241557300827
\]
\[
\text{Out[70]=} \quad 1476241557300827
\]

\[
\text{In[71]:=} \quad p_2 = 48384280209696856047731
\]
\[
\text{Out[71]=} \quad 48384280209696856047731
\]

\[
\text{In[69]:=} \quad \text{PowerMod}[2, (n - 1)/2, n]
\]
\[
\text{Out[69]=} \quad 285707540662569884530199015485751094148
\]

\[
\text{In[73]:=} \quad \text{PowerMod}[2, (n - 1)/p_1, n]
\]
\[
\text{Out[73]=} \quad 85123374597353559026878384594727525044
\]

\[
\text{In[74]:=} \quad \text{PowerMod}[2, (n - 1)/p_2, n]
\]
\[
\text{Out[74]=} \quad 212515114150837464246496688100132326772
\]

We luck out on our first try!

\(g = 2\) is a primitive root for our \(n\) by the primitive root test. So \(n\) is prime.

One could also try to find a primitive root by picking a random number.
Let's call it $a$.

Since a perfect square cannot be a primitive root we make sure our number is not a perfect square.

So $a$ is NOT a primitive root. We choose another $a$ at random.
In[85]:= 
  PowerMod[a, (n - 1) / p2, n]
Out[85]=  19528456887363581891485869954698462717

So $a = 142759089631514075232115530160417972566$ is a primitive root for our $n$ also, so we could have used this $a$ to show our $n$ was prime.

Note that the probability that a randomly chosen $a$ will be a primitive root is $\text{EulerPhi}(n - 1) / (n - 1)$. For our example:

In[90]:= 
  N[EulerPhi[n - 1] / (n - 1)]
Out[90]=  0.5

So our chances of picking a primitive root seem to be 50% !!! This is pretty good. Is it really 50%? We ask Mathematica to be more accurate.

In[91]:= 
  N[EulerPhi[n - 1] / (n - 1), 20]
Out[91]=  0.49999999999999966130

So we see that our chances wasn’t exactly 50% but pretty close. The fewer prime factors $n - 1$ has the better your chances of finding a primitive root. However with fewer prime factors of $n - 1$ it is more difficult to factor.