

Primality testing

Mike May, S. J., 1998

A simple procedure for primality testing.

In class we looked at some probabilistic tests of primality. We want to see how effective the tests are. We will simply eyeball the results to see how well the tests work.

For the first test, we test for primality by using Fermat's little theorem. We see if n is prime by taking a random number $a < n$ and seeing if $a^{n-1} = 1 \pmod n$. We have been told that if n is not prime we have at least a 50% chance of showing it by choosing a single number.

It is worthwhile checking out the effectiveness of this test with a specific n . The first time through we will look at the case of $n = 77$.

We first define a vector with all the numbers from 1 through $n-1$.

```
> alphas := linalg[vector](76,i->i);
```

```
alphas := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76]
```

Then we see what happens when we raise these numbers to the $n-1$ power mod n .

```
> betas := map(i-> [i,Power(i,76) mod 77], alphas);
```

```
betas := [[1, 1], [2, 9], [3, 25], [4, 4], [5, 16], [6, 71], [7, 70], [8, 36], [9, 9], [10, 67], [11, 11], [12, 23], [13, 64], [14, 14], [15, 15], [16, 16], [17, 60], [18, 4], [19, 58], [20, 64], [21, 56], [22, 22], [23, 23], [24, 53], [25, 25], [26, 37], [27, 71], [28, 49], [29, 15], [30, 58], [31, 53], [32, 67], [33, 44], [34, 1], [35, 42], [36, 36], [37, 37], [38, 60], [39, 60], [40, 37], [41, 36], [42, 42], [43, 1], [44, 44], [45, 67], [46, 53], [47, 58], [48, 15], [49, 49], [50, 71], [51, 37], [52, 25], [53, 53], [54, 23], [55, 22], [56, 56], [57, 64], [58, 58], [59, 4], [60, 60], [61, 16], [62, 15], [63, 14], [64, 64], [65, 23], [66, 11], [67, 67], [68, 9], [69, 36], [70, 70], [71, 71], [72, 16], [73, 4], [74, 25], [75, 9], [76, 1]]
```

Thus the test would tell us 77 is not prime if the chosen a was anything except

1, 34, 43, and 76. Thus the test indicates that n is not prime for 72 of 76 possible values of a. When we consider that the test is always inconclusive if a is 1 or -1, we see that the test indicates not prime for 72 of 74 possible a between 2 and n-2.

Test some other numbers:

The following block of code is set up to test other numbers using the first test.

```
> n := 131;
> alpha1 := linalg[vector](n-1,i->i);
> beta1 := map(i -> [i, Power(i, n-1) mod n], alpha1);
n := 131
```

$\alpha_1 := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130]$

$\beta_1 := [[1, 1], [2, 1], [3, 1], [4, 1], [5, 1], [6, 1], [7, 1], [8, 1], [9, 1], [10, 1], [11, 1], [12, 1], [13, 1], [14, 1], [15, 1], [16, 1], [17, 1], [18, 1], [19, 1], [20, 1], [21, 1], [22, 1], [23, 1], [24, 1], [25, 1], [26, 1], [27, 1], [28, 1], [29, 1], [30, 1], [31, 1], [32, 1], [33, 1], [34, 1], [35, 1], [36, 1], [37, 1], [38, 1], [39, 1], [40, 1], [41, 1], [42, 1], [43, 1], [44, 1], [45, 1], [46, 1], [47, 1], [48, 1], [49, 1], [50, 1], [51, 1], [52, 1], [53, 1], [54, 1], [55, 1], [56, 1], [57, 1], [58, 1], [59, 1], [60, 1], [61, 1], [62, 1], [63, 1], [64, 1], [65, 1], [66, 1], [67, 1], [68, 1], [69, 1], [70, 1], [71, 1], [72, 1], [73, 1], [74, 1], [75, 1], [76, 1], [77, 1], [78, 1], [79, 1], [80, 1], [81, 1], [82, 1], [83, 1], [84, 1], [85, 1], [86, 1], [87, 1], [88, 1], [89, 1], [90, 1], [91, 1], [92, 1], [93, 1], [94, 1], [95, 1], [96, 1], [97, 1], [98, 1], [99, 1], [100, 1], [101, 1], [102, 1], [103, 1], [104, 1], [105, 1], [106, 1], [107, 1], [108, 1], [109, 1], [110, 1], [111, 1], [112, 1], [113, 1], [114, 1], [115, 1], [116, 1], [117, 1], [118, 1], [119, 1], [120, 1], [121, 1], [122, 1], [123, 1], [124, 1], [125, 1], [126, 1], [127, 1], [128, 1], [129, 1], [130, 1]]$

Exercise:

1) Pick 3 nonprime numbers between 100 and 150. For each one, find out how many of the numbers between 2 and $n-2$ will show that each number is composite.

A procedure for testing higher numbers:

Obviously, we can't use this same procedure for large numbers because we will run out of memory trying to store the arrays. A simple alternative is to test with a running from 2 to 100. The procedure is given below.

A simple prime tester

```
> primetester := proc(n)
> local i, temp:
> for i from 2 to 100 do
>   temp := Power(i,n-1) mod n:
>   if temp <> 1 then
>     print(n, ' is not prime'):
>     print(i, ' ^ ', n-1, ' = ', temp):
>     break:
>   fi:od:
> if i > 100 then print(n, ' is probably prime', i) fi:
> if i > 100 then n else 0 fi:
> end:
> primetester(1122334455667789);
      1122334455667789, is not prime
      2, ^ , 1122334455667788, = , 210589670002713
      0
> primetester(131);
      131, is probably prime, 101
      131
```

Finding primes

Now that we can test if a number is prime, we can try to find a prime by testing numbers in order until we find a prime

```

> findprime := proc(n)
> local temp, temp2, a, b, c:
> temp := rand(n)():
> print('the random number is ', temp):
> a := 0:
> c:= 0:
> while a = 0 do:
> b := primetester(temp):
> temp := temp +1:
> c := c+1:
> if b <> 0 then a := 1:
> fi:od:
> print('After ',c-1,' unsuccessful tries, the prime number is
', b):
> b:
> end:

```

Let us use this procedure to find an 8 digit prime. We use the Maple command isprime to test our answer.

```

> findprime(10^8);
> isprime("");

```

Warning, incomplete string; use " to end the string

Exercise:

2) Find primes with 10, 20, 30, 40, and 50 digits.

Oops

Let us try our test on $3828001 = 101 \cdot 151 \cdot 251$ which is clearly not prime.

```

> primetester(3828001): isprime(3828001);ifactor(3828001);
3828001, is probably prime, 101
false
(101) (151) (251)

```

The number 3828001 was specially chosen so that the order of the multiplicative group divides $n-1$ even though n is not prime. Such numbers are called Carmichael numbers. We would still spot that n is not prime if we tested a number that is not relatively prime to n . Unfortunately, n was chosen so that its smallest prime factor is 101. Thus we need to use the more sensitive Miller test.

The Miller Test of primality.

We also want to see what happens with the more sophisticated test, the Miller test. For that test I factor $n-1$ as $k 2^t$ for some odd number k . We then look at a raised to the powers $k, 2*k, 2^2*k, \dots, 2^t*k = n-1$. If n is prime, then $a^{(n-1)} = 1 \pmod n$. Additionally, if $a^{(2^j)} = 1 \pmod n$, then $a^j \pmod n$ is either 1 or -1.

I notice that $76 = 2^2 19$. Thus I want to look at raising a to the powers 19, 38, and 76.

For a test to show prime, I need the last power to be 1. I also need for the previous power, if it exists, to be 1 or -1. Recall that $76 = -1 \pmod{77}$.

```
> gammas := map(i ->
> [i, Power(i,19) mod 77, Power(i,38) mod 77, Power(i,76) mod
> 77],
> alphas);

gammas := [[1, 1, 1, 1], [2, 72, 25, 9], [3, 59, 16, 25], [4, 25, 9, 4], [5, 75, 4, 16],
[6, 13, 15, 71], [7, 63, 42, 70], [8, 29, 71, 36], [9, 16, 25, 9], [10, 10, 23, 67],
[11, 11, 44, 11], [12, 12, 67, 23], [13, 6, 36, 64], [14, 70, 49, 14], [15, 36, 64, 15],
[16, 9, 4, 16], [17, 24, 37, 60], [18, 74, 9, 4], [19, 40, 60, 58], [20, 27, 36, 64],
[21, 21, 56, 56], [22, 22, 22, 22], [23, 23, 67, 23], [24, 17, 58, 53], [25, 4, 16, 25],
[26, 47, 53, 37], [27, 20, 15, 71], [28, 35, 70, 49], [29, 8, 64, 15], [30, 51, 60, 58],
[31, 38, 58, 53], [32, 32, 23, 67], [33, 33, 11, 44], [34, 34, 1, 1], [35, 28, 14, 42],
[36, 15, 71, 36], [37, 58, 53, 37], [38, 31, 37, 60], [39, 46, 37, 60], [40, 19, 53, 37],
[41, 62, 71, 36], [42, 49, 14, 42], [43, 43, 1, 1], [44, 44, 11, 44], [45, 45, 23, 67],
[46, 39, 58, 53], [47, 26, 60, 58], [48, 69, 64, 15], [49, 42, 70, 49], [50, 57, 15, 71],
[51, 30, 53, 37], [52, 73, 16, 25], [53, 60, 58, 53], [54, 54, 67, 23], [55, 55, 22, 22],
[56, 56, 56, 56], [57, 50, 36, 64], [58, 37, 60, 58], [59, 3, 9, 4], [60, 53, 37, 60],
[61, 68, 4, 16], [62, 41, 64, 15], [63, 7, 49, 14], [64, 71, 36, 64], [65, 65, 67, 23],
[66, 66, 44, 11], [67, 67, 23, 67], [68, 61, 25, 9], [69, 48, 71, 36], [70, 14, 42, 70],
[71, 64, 15, 71], [72, 2, 4, 16], [73, 52, 9, 4], [74, 18, 16, 25], [75, 5, 25, 9],
[76, 76, 1, 1]]
```

Thus, the test shows 77 is not a prime unless we choose a to be 1 or -1. In effect the test shows that n is not prime if we choose any a between 2 and $n-2$.

To do the Miller test we first need to express $n-1$ as an odd number times a power of 2. We do that with the procedure `oddfactor`. The procedure `primetester2` below performs the Miller test and look for number that the Miller fail to witness to the fact that n is not prime.

```

> oddfactor := proc(n)
> local temp, count:
> temp := n:
> count := 0:
> while (temp mod 2) = 0 do
> temp := temp/2:
> count := count + 1:
> od:
> [temp, count];
> end:
> primetester2 := proc(n)
> local i, temp, nums, count, temp2:
> temp2 := n:
> nums := oddfactor(n-1):
> for i from 2 to 100 do
> temp := Power(i,n-1) mod n:
> if temp <> 1 then
> if temp2 <> 0 then
> print(n, ' is not prime'):
> print(i, '^ ', n-1, ' = ', temp):
> fi:
> temp2 := 0:
> break:
> else
> for count from 1 to nums[2] do
> temp := Power(i, (n-1)/2^count) mod n:
> if (temp <> 1) and ((n-temp) <> 1) then
> if temp2 <> 0 then
> print(n, ' is not prime'):
> ### WARNING: note that 'I' is no longer of type '^'
> print(i, '^ ', (n-1)/2^count, ' = ', temp, ' and ',
> ### WARNING: note that 'I' is no longer of type '^'
> i, '^ ', (n-1)/2^(count-1), '=1'):
> fi:
> temp2 := 0:
> break:
> fi:
> if temp = -1 then
> print(i, '^ ', (n-1)/2^count, ' = ', temp):
> break:
> fi:
> od:
> if count > nums[2] then print(i, ' is not a witness'):fi
> fi:
> od:
> if temp2 = n then print(n, ' is probably prime', i) fi:
> temp2;
> end:
> primetester2(3828001);

```

3828001, *is not prime*

2, **proc()** option *builtin*; 85 **end proc**, 239250, = , 1174932, and , 2,
proc() option *builtin*; 85 **end proc**, 478500, = 1

5, *is not a witness*
 6, *is not a witness*
 14, *is not a witness*
 16, *is not a witness*
 25, *is not a witness*
 30, *is not a witness*
 31, *is not a witness*
 33, *is not a witness*
 36, *is not a witness*
 58, *is not a witness*
 68, *is not a witness*
 70, *is not a witness*
 77, *is not a witness*
 80, *is not a witness*
 81, *is not a witness*
 82, *is not a witness*
 84, *is not a witness*
 88, *is not a witness*
 96, *is not a witness*
 0

Note that only 18 of the numbers from 2 through 100 fail to witness that n is not prime.

Compare the results to a few other Carmichael numbers. With two Carmichael numbers we note that the first primality test fails to identify them as composite numbers, then note that the Miller test does identify them as composites, then we factor the numbers.

```

> primetester(6733693):    primetester2(6733693):    ifactor(6733693);
> primetester(34657141): primetester2(34657141): ifactor(34657141);

```

6733693, *is probably prime*, 101

6733693, *is not prime*

```

2, proc() option builtin; 85 end proc, 3366846, = , 2594635, and , 2,
proc() option builtin; 85 end proc, 6733692, = 1

```

9, *is not a witness*

12, *is not a witness*

16, *is not a witness*

21, *is not a witness*
22, *is not a witness*
25, *is not a witness*
26, *is not a witness*
28, *is not a witness*
29, *is not a witness*
31, *is not a witness*
49, *is not a witness*
81, *is not a witness*
82, *is not a witness*
97, *is not a witness*
(109) (163) (379)

34657141, *is probably prime*, 101

34657141, *is not prime*

2, **proc() option builtin; 85 end proc**, 17328570, = , 15640991, *and* , 2,
proc() option builtin; 85 end proc, 34657140, = 1

3, *is not a witness*
7, *is not a witness*
9, *is not a witness*
16, *is not a witness*
20, *is not a witness*
21, *is not a witness*
25, *is not a witness*
27, *is not a witness*
46, *is not a witness*
48, *is not a witness*
49, *is not a witness*
60, *is not a witness*
63, *is not a witness*
74, *is not a witness*
75, *is not a witness*
81, *is not a witness*
94, *is not a witness*
97, *is not a witness*
(191) (421) (431)

Exercise:

3) Pick 3 random 50 digit numbers. Test if each to see if it is prime and if it is composite, note the percentage of choices of a from 2 to 100 that will show they are composite with the Miller test.

It is worth noting that we have yet to find a composite n that we can't show to be composite by testing the number 2. Such a number would be called a strong probable-prime base 2. It has been shown in the literature that letting a range over the primes less than 20 will correctly identify all composites less than 10^{15} .

We can clean up the code and fine a better prime finding routine

```

> oddfactor := proc(n)
> local temp, count:
> temp := n:
> count := 0:
> while (temp mod 2) = 0 do
> temp := temp/2:
> count := count + 1:
> od:
> [temp, count];
> end:
> primetester3 := proc(n)
> local i, temp, nums, count, temp2:
> temp2 := n:
> nums := oddfactor(n-1):
> for i from 2 to 100 do
> temp := Power(i,n-1) mod n:
> if temp <> 1 then
> if temp2 <> 0 then
> print(n, ' is not prime'):
> print(i, '^ ', n-1, ' = ', temp):
> fi:
> temp2 := 0:
> break:
> else
> for count from 1 to nums[2] do
> temp := Power(i, (n-1)/2^count) mod n:
> if (temp <> 1) and ((n-temp) <> 1) then
> if temp2 <> 0 then
> print(n, ' is not prime'):
> ### WARNING: note that 'I' is no longer of type '^'
> print(i, '^ ', (n-1)/2^count, ' = ', temp, ' and ',
> ### WARNING: note that 'I' is no longer of type '^'
> i, '^ ', (n-1)/2^(count-1), '=1'):
> fi:
> temp2 := 0:
> break:
> fi:
> if temp = -1 then
> print(i, '^ ', (n-1)/2^count, ' = ', temp):
> break:
> fi:
> od:
> fi:
> if temp2 = 0 then break: fi:
> od:
> if temp2 = n then print(n, ' is probably prime', i) fi:
> temp2;
> end:
> findprime2 := proc(n)
> local temp, temp2, a, b, c:
> temp := rand(n)():
> print('the random number is ', temp):
> a := 0:
> c := 0:
> while a = 0 do
> b := primetester3(temp):
> temp := temp + 1:
> c := c + 1:
> if b <> 0 then a := 1: fi:
> od:
> print('After ', c-1, ' unsuccessful tries, the prime number is
', b):
> b:
> end:

```

```
> findprime2(10^20); isprime("");
```

Warning, incomplete string; use " to end the string

Exercise:

4) Use the findprime2 procedure to find 60, 70, and 80 digit primes.