Solutions to Project Euler Problem 4: Largest Palindrome Product

Alister McKinley

https://projecteuler.net/problem=4

The problem reads:

> A palindromic number reads the same both ways. The largest palindrome made from the product of two 2-digit numbers is 9009 = 91 × 99.
>
> Find the largest palindrome made from the product of two 3-digit numbers.

Brute Force Approach

The most straight forward (and least efficient) way of solving this is to consider every product of two three-digit numbers, x and y, locate the palindromes, and retain the largest one.

We start by setting both factors x and y to 100, then we increment y while looking for palindromes. Once we have exhausted the values for y, we increase x by 1 and repeat the process with y.

x	y	xy	Palindrome?	Largest Found
100	100	10000	False	
100	101	10100	False	
...	...	...	...	...
100	999	99900	False	
101	100	10100	False	
101	101	10201	True	10201
...	...	...	...	...
913	993	906609	True	906609
...	...	...	...	...

Both x and y can take 900 different values (100 to 999), leading to $900^2 = 810000$ results to check. This method will eventually identify $913 \cdot 993 = 906609$ as the answer.

A More Efficient Way

Now we will attempt to find a more efficient way to solve the problem. The first thing to notice is that the problem is asking for the *largest* palindrome. This means we should reverse the order of our search by starting with large values for our factors and moving down to smaller values.

We also keep track of the largest palindrome found. As soon as the product we are considering is smaller than the largest palindrome found we can skip the remaining y values for that x value. This is because we are decreasing the value of y so the resulting products will all be smaller. This reduces the number of products that will be checked down to 9201.

x	y	xy	Palindrome?	Largest Found
999	999	998001	False	
999	998	997002	False	
...	...	...	...	...
999	100	99900	False	
998	999	997002	False	
...	...	...	...	...
996	100	99600	False	
995	999	994005	False	
...	...	...	...	...
995	583	580085	True	580085
994	999	993006	False	580085
...	...	...	...	...
994	584	580496	False	580085
994	583	579502	False	580085
993	999	992007	False	580085
...	...	...	...	...
993	913	906609	True	906609
...	...	...	...	...

The next thing to notice is that integer multiplication is commutative, meaning that changing the order the factors are multiplied in does not change the product e.g. $2 \cdot 3 = 3 \cdot 2 = 6$.

When we decrease the value of x, instead of starting y at 999 we will start it at the same value of x. This is because the products with larger values of y have already been tested when x had those values. The table above shows this, we have both $999 \cdot 998 = 997002$ and $998 \cdot 999 = 997002$ being tested. This change reduces the number of products that will be checked down to 7020. The

table now looks like:

x	y	xy	Palindrome?	Largest Found
999	999	998001	False	
999	998	997002	False	
...	...	...	...	...
999	100	99900	False	
998	998	996004	False	
...	...	...	...	...
996	100	99600	False	
995	995	990025	False	
...	...	...	...	...
995	583	580085	True	580085
994	994	988036	False	580085
...	...	...	...	...
994	584	580496	False	580085
994	583	579502	False	580085
993	993	986049	False	580085
...	...	...	...	...
993	913	906609	True	906609
...	...	...	...	...

Now we will do some analysis on the palindrome itself. We've already seen a six-digit palindrome, 580085, and we know that the largest product we will consider is $999^2 = 998001$, therefore we know the largest palindrome will have six digits. We will call our palindrome P and write its digits as:

$$P = \overline{abccba}$$

where:

a, b, c, c, b, a	are the digits of P, in that order
$a \in \{1, \ldots, 9\}$	leading digit of P
$b, c \in \{0, \ldots, 9\}$	non-leading digits of P

Using the above definitions we see:

$$\begin{aligned} P &= \overline{abccba} \\ &= 100000a + 10000b + 1000c + 100c + 10b + a \\ &= 100001a + 10010b + 1100c \\ &= (11 \cdot 9091)a + (2 \cdot 5 \cdot 7 \cdot 11 \cdot 13)b + (2^2 \cdot 5^2 \cdot 11)c \\ &= 11 \cdot (9091a + 910b + 100c) \end{aligned}$$

This clearly shows that the product P is a multiple of 11. Since 11 is a prime number, it can't be broken up into smaller prime factors, and so it must be a prime factor of x or y (or both).

The fact that P is a multiple of 11 can also be shown with a divisibility test for 11. If the alternating sum of the digits of a number is equal to 0, then the number is a multiple of 11:

$$S_{\pm}(n) = 0 \iff n \equiv 0 \pmod{11}$$

where:

$S_{\pm}(n)$ is the alternating digit sum of n

Applied to P:

$$\begin{aligned} P &= \overline{abccba} \\ S_{\pm}(P) &= S_{\pm}(\overline{abccba}) \\ &= a - b + c - c + b - a \\ &= (a - a) + (b - b) + (c - c) \\ &= 0 \end{aligned}$$

Since we only try y values that are smaller than or equal to x we will assume y is the multiple of 11 so that no eligible products are skipped. This reduces the number of products that will be checked down to 1439. The table now looks like:

x	y	xy	Palindrome?	Largest Found
999	990	989010	False	
999	976	978021	False	
...	...	...	...	...
999	110	109890	False	
998	990	988020	False	
...	...	...	...	...
996	110	109560	False	
995	990	985050	False	
...	...	...	...	...
995	583	580085	True	580085
994	990	984060	False	580085
...	...	...	...	...
994	594	590436	False	580085
994	583	579502	False	580085
993	990	986049	False	580085
...	...	...	...	...
993	913	906609	True	906609
...	...	...	...	...

There are further improvements that can be made, but the complexity of the solution increases for only a marginal gain. We have already reduced the number of products checked from 810000 down to 1439.

Python Implementation

```
max_palindrome = 0
for x in range(999, 99, -1):
    for y in range(11*floor(x/11.), 99, -11):
        product = x * y
        if product < max_palindrome: break
        if str(product) == str(product)[::-1]:
            if product > max_palindrome:
                max_palindrome = product

print(max_palindrome)
```