Perl Weekly Challenge - 022, Task #1 • Mark Senn

Created: 19-08-24 18:54 +00 • Last Revised: 2019-08-25 17:07 +00

From Perl Weekly Challenge - 022, Task #1 retrieved on 2019-08-22 at 14:45 +00:

> Write a script to print first **10 Sexy Prime Pairs**. Sexy primes are prime numbers that differ from each other by 6. For example, the numbers 5 and 11 are both sexy primes, because $11 - 5 = 6$. The term "sexy prime" is a pun stemming from the Latin word for six: sex. For more information, please checkout wiki page.

Discussion

I'm using "6-prime(s)" instead of "sexy prime(s)" because it is shorter and more mathematicially descriptive. This notation is easier to generalize to primes that differ by eight (8-primes), etc.

There are *lots* of ways to solve this problem. I decided to try the following solution.

Inside a loop, compute a fresh prime p, if the prime $p - 6$ is in the circular buffer the program maintains, print the 6-primepair $(p - 6, p)$. Put p in the circular buffer and repeat the loop.

I usually don't like to optimize programs because human time is worth more than computer time. And, as Don Knuth stated "Premature optimization is the root of all evil." But, figuring out the minimal circular buffer size sounds like fun and I don't know any pithy sayings about minimizing human fun offhand.

What is the minimum size of the circular buffer to find all 6-prime pairs?

Let p_1, p_2, ... be the prime numbers. The first 6-prime pair is $(p_3, p_5) = (5, 11)$.

The table below is used in the discussion immediately following the table. "%%n" means that the number for this row is evenly divisible by n.

1	**2**	**3**	**4**	**5**
p_r	prime			
$p_r - 1$		%%2	%%3	
$p_r - 2$				%%3
$p_r - 3$		%%2		
$p_r - 4$			%%3	
$p_r - 5$		%%2		%%3
$p_r - 6$	prime			

Column 1: Let r be a random, positive integer, that's greater than five. Let $(p_r - 6, p_r)$ be a pair of 6-primes. I suspect this algorithm will also find the first pair of 6-primes, running the program will test that.

Column 2: Both $p_r - 6$ and p_r must be prime.

Column 3: If p_r is prime when $r > 5$ then p_r must be odd so $p_r - 1$, $p_r - 3$, and $p_r - 5$ must be evenly divisible by 2.

Column 4: Either column 4 or column 5 are true but not both. If $p_r - 1$ and $p_r - 4$ are %%3 then $p_r - 2$ might be prime.

Column 5: Either column 4 or column 5 are true but not both. If $p_r - 2$ and $p_r - 5$ are %%3 then $p_r - 4$ might be prime.

The circular buffer must contain two elements. One element will not be used (if there is no prime between $p_r - 6$ and p_r), will hold a prime (see Column 4 case above), or will hold a prime (see Column 5 case above). The other element will hold $p_r - 6$.

Perl 6 solution

```
#
# Perl Weekly Challenge - 022
# Task #1
#
# See
#     engineering.purdue.edu/~mark/pwc-022-1.pdf
# for more information.

# Run using Perl v6.d.
use v6.d;

my $d    =  6;    # sexy primes differ by 6
my $n    = 10;    # print the first $n sexy prime pairs
my $p    =  0;    # how many sexy prime pairs have been printed
my $size =  2;    # size of the circular buffer

# Make a $size element circular buffer.
# Make each element the illegal value of 2 - $d - 1 (= -5).
# The circular buffer will get seeded with the correct values later.
# The first sexy prime pair will be found at (5,11).
my @cb[$size] = (2-$d-1) xx *;

my $i = 0;                                # Current index into cb
for ((2..Inf).grep({.is-prime})) {        # For each prime number...
    if ($_-$d == @cb.any) {               #   is it part of a sexy prime pair?
        "({$_-$d},$_)".say;               #     print the sexy prime pair
        (++$p == $n)  and  last;          #     have we printed $n sexy prime pairs?
    }
    @cb[$i++ % $size] = $_;               # put the new prime in the circlar buffer
    # say "cb = {@cb}";                   # (for testing)
}
```

History

2019-08-24	Finished first version.
2019-08-25	Changed “sexy prime” to “6-prime”.
	Changed “//” to “%%” for “evenly divisible”.
	Center table headings.