Pentagonal Numbers • Mark Senn • last updated on 2022-01-15 at 21:19-05

Problem Statement

From The Weekly Challenge - 147 Task #2: Pentagon Numbers retrieved on 2022-01-15 at 12:23-05:

> **Submitted by:** Mohammad S Anwar
>
> Write a script to find the first pair of Pentagon Numbers whose sum and difference are also a Pentagon Number.
>
> > Pentagon numbers can be defined as P(n) = n(3n -1)/2.
>
> **Example**
>
> ```
> The first 10 Pentagon Numbers are:
> 1, 5, 12, 22, 35, 51, 70, 92, 117 and 145.
>
> P(4) + P(7) = 22 + 70 = 92 = P(8)
> but
> P(4) - P(7) - |22 - 70| = 48 is not a Pentagon Number.
> ```

Pentagonal number math summary

The following is based on information from Wikipedia's Pentagonal number page.

The first few pentagonal numbers are $p_1,\ p_2,\ \ldots = 1,\ 5,\ 12,\ 22,\ 35,\ \ldots$.

p_n is given by the formula:

$$p_n = \frac{3n^2 - n}{2} \tag{1}$$

for $n \geq 1$.

The recurrence relation

$$p_n = 2p_{n-1} - p_{n-2} + 3 \tag{2}$$

computes p_n from p_{n-1} and p_{n-2}.

Given a positive integer x, to test whether it is a pentagonal number compute

$$n = \frac{\sqrt{24x+1}+1}{6}. \tag{3}$$

The number x is pentagonal if and only if n is a natural number. In that case x is the nth pentagonal number.

Raku Solution

```
# Use the first n pentagonal numbers.  The zeroth element is not used so
# p_1, p_2, ..., p_n will correspond to $p[1], $p[2], ..., $p[n].
#
# The maximum pair of pentagonal numbers that can be tested are p_i and p_j
# where 3 <= i < n, 2 <= j < i, and p_i + p_j <= p_n.
my $n = 1_000_000;
```

```
# Use Raku's sequence operator to define an infinite lazy list containing
# the pentagonal numbers.  See
#     https://perl6advent.wordpress.com/2010/12/04/the-sequence-operator
# for more information on the sequence operator.  $p[0] (= 0) is not used.
my $p := 0, 1, 5, -> $a, $b { 2*$b - $a + 3 } ... Inf;

# Make a is-pentagonal hash where %is-pentagonal{p_i} is true for the first
# n pentagonal numbers.
my %is-pentagonal;
(1 .. $n).map({%is-pentagonal{$p[$_]} = True});

for (3 .. $n-1) -> $i  {
    for (2 .. $i-1) -> $j  {

        # Is p_i + p_j a pentagonal number?
        my $t = $p[$i] + $p[$j];
        ($t > $p[$n])  and  die "sum $t is too big to test";
        %is-pentagonal{$t}  or  next;

        # Is p_i - p_j a pentagonal number?
        $t = $p[$i] - $p[$j];
        %is-pentagonal{$t}  or  next;

        # The following numbers get printed: 1560090 and 7042750.
        print qq:to/END/;
        The first pair of pentagonal numbers that satisfy
        the conditions is ($p[$j], $p[$i]).
        END
        exit 0;

    }
}
```