ERIC BAILEY

ABSTRACT ALGEBRA IN GAP

Contents

Basic System Interaction

Exercise 1

a IsPerfectInt is a function that takes a positive integer n and returns true if n is perfect and false otherwise.

We could define a function to compute the aliquot sum of a positive integer n:

5a ⟨*Compute the aliquot sum of a positive integer* 5a⟩≡

```
AliquotSum := n -> Sum(DivisorsInt(n)) - n;
```

$$s(n) \equiv \sigma(n) - n$$

Defines:
AliquotSum, used in chunk 5b.

Then, using that definition, we could write a function to determine whether a positive integer n is perfect:

5b ⟨*Determine whether a positive integer is perfect* 5b⟩≡

```
IsPerfectInt := n -> n = AliquotSum(n);
```

Uses AliquotSum 5a and IsPerfectInt 5c.

Conveniently, GAP ships with Sigma, which we can use instead.

$$\sigma(n) = \sum_{d|n} d$$

5c ⟨*Determine whether a positive integer is perfect, using Sigma* 5c⟩≡ (6c)

```
IsPerfectInt := n -> Sigma(n) = 2*n;
```

$$\texttt{IsPerfectInt}(n) := \sigma(n) = 2n$$

Defines:
IsPerfectInt, used in chunks 5 and 6.

b To find all perfect numbers less than 1000, run the following:

5d ⟨*Find all perfect numbers less than 1000* 5d⟩≡ (6d)

```
Filtered([1..999], IsPerfectInt);
```

$$\{n \in \mathbb{Z}^+ \mid 1 \le n < 1000,\ \texttt{IsPerfectInt}(n)\}$$

Uses IsPerfectInt 5c.

... which results in:

5e ⟨*All perfect numbers less than 1000* 5e⟩≡ (6d)

```
[ 6, 28, 496 ]
```

c Not all numbers of the form $2^n(2^{n+1} - 1)$, for some positive integer n, are perfect.

6a ⟨*Not all perfect* 6a⟩≡
```
gap> ForAll( PositiveIntegers,
>            n -> IsPerfectInt(2^n * (2^(n+1) - 1)) );
false
```
Uses IsPerfectInt 5c.

d In Euclid's formation rule (IX.36), he proved $\frac{q(q+1)}{2}$ is an even perfect number where q is a prime of the form $2^p - 1$ for prime p, a.k.a. a Mersenne prime.

6b ⟨*Euclid's IX.36* 6b⟩≡
```
gap> MersennePrimes := Filtered( List( Primes{[1..50]},
                                       p -> 2^p - 1 ),
                                 IsPrime );
[ 3, 7, 31, 127, 8191, 131071, 524287, 2147483647,
  2305843009213693951, 618970019642690137449562111,
  162259276829213363391578010288127,
  170141183460469231731687303715884105727 ]
gap> ForAll( MersennePrimes, q -> IsPerfectInt(q * (q + 1) / 2) );
true
```
Uses IsPerfectInt 5c.

e TODO: Prove it.

Code

6c ⟨*src/PerfectNumbers.g* 6c⟩≡
⟨*Determine whether a positive integer is perfect, using Sigma* 5c⟩

Tests

To run the tests, make sure the code is loaded (`Read("./src/PerfectNumbers.g");`), then run `Test("src/PerfectNumbers.tst");`.

6d ⟨*src/PerfectNumbers.tst* 6d⟩≡
```
# Perfect Number Tests

# Perfect numbers less than 1000
gap> ⟨Find all perfect numbers less than 1000 5d⟩
⟨All perfect numbers less than 1000 5e⟩
```

Chunks

Index

Bibliography