Artem Bolshakov

Commentary on Python Modules

The attached python modules were intended for work with primitive abundant numbers, as described in the main text (abundant or perfect numbers with no abundant or perfect positive proper divisors). However, due to both inexperience with computing (hence the multiple functions that do the same thing, eg "primes" and "eratosthenes" in primes.py) and due to the more general number theoretic objectives tangent to the subject (hence the functions in number_theory.py that have nothing to do with primitive abundant numbers, like "tau"), the size and scope of these modules has increased. While this is good, as this process has created modules more generally applicable than merely to this publication, this also makes the job for someone who wants to check the validity of these functions more troublesome. Therefore, here is a list of functions important for this subject, by module:

calc2: entire module
calc3: entire module
calc4: entire module
primab2: entire module
number_theory2: LCM, nu, GCD, sigma, isabundant, isperfect, isdeficient
primes2: eratosthenes, pf, lim, upperodd, exponent, divtest, mult, permult, factors, propdiv

With these functions, it is possible to look for patterns and otherwise generally experiment with primitive abundant numbers, and use the computational power of the computer.

Now, more specifically, the computer was initially intended to compute the computationally intensive question of the natural density of abundant numbers, a question to which primitive abundant numbers lend themselves perfectly, because all we need to find is the natural density of numbers divisible by at least one of these. The python module "calc" was specifically created for this purpose. The functions (by module) that are necessary for that purpose are below:

calc2: entire module
calc3: entire module
calc4: entire module
primab2: primab
number_theory2: LCM, GCD, nu, sigma
primes2: eratosthenes, pf, lim, upperodd, exponent, divtest

The following function call in any python-running window will calculate the natural density of all the abundant numbers divisible by all the primitive abundant numbers less than or equal to n:

```
>>>import primab2
>>>import calc2[1]
>>>calc2.probdiv(primab2.primab(n))
```

For instance, the above function call for all the primitive abundant numbers below 1000 yields approximately 0.2402372. Unfortunately, this function takes exponential time with respect to the quantity of primitive abundant numbers, so I was unable to obtain original results on the machines available to me; however, this function, given sufficient memory, will return arbitrarily good lower bounds on the natural density of primitive abundant numbers.

Attached is separate commentary on the subject of computing the lower bound for the natural density of abundant numbers, which is a question sufficiently complex to deserve its own commentary. So far, the best number obtained was ~ 0.2427, obtained from all the primitive abundant numbers less than 3000.

1 or 3, or 4