Problem x.yz. Let d(n) be defined as the sum of proper divisors of n (numbers less than n which divide evenly into n). If d(a) = b and d(b) = a, where $a \neq b$, then a and b are an amicable pair and each of a and b are called amicable numbers. For example, the proper divisors of 220 are 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 and 110; therefore d(220) = 284. The proper divisors of 284 are 1, 2, 4, 71 and 142; so d(284) = 220. Evaluate the sum of all the amicable numbers under 10000.

Knowledge Required How to find divisors in $O(\sqrt{n})$ time.

Solution Outline We start off by implementing a function called `D` which takes in a number and returns the sum of proper divisors of that number. Then we start iterating numbers from 1 until 10000 and check if `D(i) = D(D(i))` if it satisfies the condition we add the number to the final answer. Note that we can speed this up by calculating `D(i)` once and storing it in a `temp` variable and then using it as `temp = D(temp)` instead of `D(i) == D(D(i))`. Final time complexity $O(n\sqrt{n})$.

Python Solution

```python
def D(n):
    div_sum = 1
    i = 2
    while i * i <= n:
        if n % i == 0:
            div_sum += i
            # if they are the same for ex n = 36, and 6 6
            if n // i != i:
                div_sum += n // i
        i += 1

    return div_sum

i = 1
amic_sum = 0
while i < 10000:
    temp = D(i)
    if i != temp and i == D(temp):
        amic_sum += i

    i += 1

print(amic_sum)
```