Daily Coding Problem

Blog

Daily Coding Problem #184

Problem

This problem was asked by Amazon.

Given n numbers, find the greatest common denominator between them.

For example, given the numbers `[42, 56, 14]`, return `14`.

Solution

Because the greatest common divisor is associative, `gcd` of multiple numbers: say, `a, b, c` is equivalent to `gcd(gcd(a, b), c)`. Intuitively, this is because if `d` divides `gcd(a, b)` and `c`, it must divide `a` and `b` as well by the definition of the greatest common divisor.

Thus the greatest common divisor of multiple numbers `a, b, ..., z` can be obtained by iteratively computing the `gcd` of `a` and `b`, and `gcd` of the result of that with `c`, and so on.

```
def gcd(nums):
    n = nums[0]
    for num in nums[1:]:
        n = _gcd(n, num)
    return n
```

How to implement `_gcd()` though? A naive implementation might try every integer from 1 to `min(a, b)` and see if it divides the larger:

```
def _gcd_naive(a, b):
    smaller, larger = min(a, b), max(a, b)
    for d in range(smaller, 0, -1):
        if larger % d == 0:
            return d
```

However, a much more efficient method is the Euclidean algorithm to find the the greatest common divisor, which follows the recursive formula:

```
gcd(a, 0) = a
gcd(a, b) = gcd(b, a % b)
```

```python
def _gcd(a, b):
    if b == 0:
        return a
    return _gcd(b, a & b)
```

A more memory-efficient method that works bottom up:

```python
def _gcd(a, b):
    while b:
        a, b = b, a % b
    return a
```