Deacon Steiner COSC 3020

Lab 2 Explanation

METHOD:

I immediately recognized this as a greatest common divisor problem, I also knew from class that Euclid's algorithm was the most optimal way to calculate.

APPROACH:

I quickly implemented a Euclid(a, b) helper function that would return the gcd of its two parameters. Then from there i realized that to calculate number of squares needed to tile was just [(length / gcd) * (height / gcd)]. I actually went back and used floor division because the side length for each square should always be an integer based on the previous math.

REASONING:

I knew that Euclid was optimal for gcd problems because that's what we were told in class. But in that same class he told us not to just accept things implicitly, so I went down a rabbit hole on gcd optimization algorithms and found that Euclid is in fact optimal at this scale because the only improvements are constant factors, and are really only more optimal on the very upper range (hundreds-thousands of digits) So instead of trying to fight with a sub-optimal (for this assignment) algorithm I just stuck with Euclid

REFLECTION:

I used three test cases:

length: 168 and height: 64,
length: 15 and height: 20,
and length: 5 and height: 5

I wanted to check against the examples given in the lab doc, as well as verify that my solution worked when W==H (it does).