

# Homework 1

## 1. Section 1.1 #6

- (a) We find  $\gcd(31415, 14142)$  by Euclid's Algorithm.

$$\begin{aligned}
 \gcd(31415, 14142) &= \gcd(14142, 3131) \\
 &= \gcd(3131, 1618) \\
 &= \gcd(1618, 1513) \\
 &= \gcd(1513, 105) \\
 &= \gcd(105, 43) \\
 &= \gcd(43, 19) \\
 &= \gcd(19, 5) \\
 &= \gcd(5, 4) \\
 &= \gcd(4, 1) \\
 &= \gcd(1, 0) \\
 &= 1
 \end{aligned}$$

- (b) We see that the number of iterations for Euclid's algorithm is 11. The number of iterations for the method of consecutive integers is 14142, or approximately 1286 times faster.

Writing a quick Python program to run one million simulations reveals that, on average, Euclid's algorithm takes about 11 iterations and the consecutive integers method takes about 39960 iterations, or about 3633 times faster.

## 2. Section 1.1 #8

Let  $m, n \in \mathbb{Z}$  with  $m < n$ . Applying Euclid's algorithm, we have  $\gcd(m, n) = \gcd(n, m \bmod n)$ . But since  $m < n$ , then  $m \bmod n = m$ , so  $\gcd(n, m \bmod n) = \gcd(n, m)$ . In other words, the two inputs are swapped.

The inputs swapping can only happen on the first iteration and if  $m < n$ . For any integers  $a, b$ ,  $a > b \bmod a$ , since  $b \bmod a$  is the remainder of  $b$  when divided by  $a$  and the remainder is smaller than the quotient, by the Division Algorithm.

**3.** Section 1.2 #3

Only (a) is an algorithm (provided we know how to compute a square root). The problems with (b) and (c) is that they are ambiguous, i.e we are not given how to compute  $A$  or  $h_a$  respectively.

**4.** Section 1.2 #5

- (a) Let  $n$  be a positive integer and let  $r = n - 2a$ , where  $a$  is a positive integer. Then prepend  $r$  to the binary representation of  $n$ . Then assign  $n$  to  $a$  and repeat this process until  $n = 0$ .
- (b)

**5.** Section 1.3 #4

- (a) Let the two islands and the two banks of the river be the vertices and the bridges be the edges of a graph. Now we reformulate the question in graph theoretic terms: does this graph have an Euler cycle?
- (b)

**6.** Section 1.3 #8**7.** Section 2.1 #1**8.** Section 2.1 #2**9.** Section 2.1 #4**Bonus.** Section 1.2 #2