Correctness and Cost Analysis of Euclid’s Algorithm

Enric Rodríguez

Here we will study Euclid’s algorithm which, given two natural numbers \( a \) and \( b \), computes its greatest common divisor \( \text{GCD}(a,b) \). Let us consider this version:

```c
int euclid (int a, int b) {
    /* Pre:  \( a \geq b \) \( b \geq 0 \) */
    /* Post: \( \text{euclid}(a,b) = \text{GCD}(a,b) \) */
    if (b == 0) return a;
    else return euclid(b, a%b);
}
```

Let us prove its correctness by induction over \( N \), the number of recursive calls:

- **Base case**: \( N = 0 \). In this case it must be \( b = 0 \), and therefore \( \text{GCD}(a,b) = \text{GCD}(a,0) = a \). So the algorithm returns the right value.

- **Inductive case**: Let \( a \) and \( b \) be such that \( \text{euclid}(a,b) \) makes \( N \) recursive calls, with \( N > 0 \). Let \( q \) and \( r \) be the quotient and the remainder of dividing \( a \) into \( b \), so that \( a = qb + r \) and \( 0 \leq r < b \). As \( r = a\%b \), the precondition of the recursive call \( \text{euclid}(b,a\%b) \) holds. Further, \( \text{GCD}(a,b) = \text{GCD}(b,a) = \text{GCD}(b, qb + r) = \text{GCD}(b, r) \). So by IH the algorithm returns the right value.

Next let us analyze the cost of the algorithm. Except for the recursive call, the work to do has constant cost. Hence, the cost is proportional to the number of recursive calls. So we will focus on studying, given \( a \) and \( b \), which is the number of recursive calls of \( \text{euclid}(a,b) \).

Let \( F_N \) be the Fibonacci sequence: \( F_0 = 1 \), \( F_1 = 1 \) i \( F_N = F_{N-1} + F_{N-2} \) for \( N \geq 2 \). Let us show by induction that if for \( a \) and \( b \) the algorithm makes \( N \) recursive calls (where \( N \geq 1 \)), then \( a \geq F_{N+1} \) i \( b \geq F_N \):

- **Base case**: \( N = 1 \). If a single recursive call is made, then \( b \neq 0 \). As \( b \geq 0 \) by the precondition, it must be \( b \geq 1 = F_1 \). Moreover, as the precondition also ensures \( a > b \), we have \( a \geq 2 = F_2 \).

- **Inductive case**: \( N > 1 \). Let \( a \) and \( b \) be such that \( \text{euclid}(a,b) \) makes \( N \) recursive calls. Let also \( q \) and \( r \) be the quotient and the remainder of dividing \( a \) into \( b \), so that \( a = qb + r \) and \( 0 \leq r < b \). Then \( \text{euclid}(b,r) \) requires \( N - 1 \) recursive calls. By IH, \( b \geq F_N \) i \( r \geq F_{N-1} \). Furthermore, since \( a > b \), it must be \( q \geq 1 \). Hence, \( a \geq b + r \geq F_N + F_{N-1} = F_{N+1} \).

So we have proved that, if \( \text{euclid}(a,b) \) makes \( N \) recursive calls, then \( a \geq F_{N+1} \) and \( b \geq F_N \). Moreover, it can be proved (see below) that for all \( N \geq 0 \) it holds that \( F_N \geq \phi^{N-1} \), where \( \phi = \frac{1+\sqrt{5}}{2} \approx 1.618 \) is called the golden ratio. Using this, we have that if for \( a \) and \( b \) Euclid’s algorithm makes \( N \) recursive calls, then \( b \geq F_N \geq \phi^{N-1} \), whence \( \log_{\phi} b \geq N - 1 \) and \( 1 + \log_{\phi} b \geq N \). So \( N \) is \( \mathcal{O}(\log b) \).

Finally, let us prove by induction that for all \( N \geq 0 \) we have \( F_N \geq \phi^{N-1} \):

- **Base case 1**: For \( N = 0 \), we have \( F_0 = 1 > \phi^{-1} (\phi^{-1} \approx 0.618) \).
- **Base case 2**: For \( N = 1 \), we have \( F_1 = 1 = \phi^0 \).
- **Inductive case**: Let us assume \( N \geq 2 \). Notice that \( \phi^2 = (\frac{1+\sqrt{5}}{2})^2 = \frac{3 + \sqrt{5}}{4} = \phi + 1 \). So, by IH, \( F_N = F_{N-1} + F_{N-2} \geq \phi^{N-2} + \phi^{N-3} = \phi^{N-3}(\phi + 1) = \phi^{N-1} \).