b and r_1, it must divide r_2, and so on, until you finally conclude that it must divide the last nonzero remainder. On the other hand, working from the last row up, one quickly sees that the last remainder must divide all of the previous remainders and also a and b. Thus, it is the g.c.d., because the g.c.d. is the only number which divides both a and b and at the same time is divisible by any other number which divides a and b.

We next prove the time estimate. The main question that must be resolved is how many divisions we're performing. We claim that the remainders are not only decreasing, but they're decreasing rather rapidly. More precisely:

Claim. $r_{j+2} < \frac{1}{2}r_j$.

Proof of claim. First, if $r_{j+1} \le \frac{1}{2}r_j$, then immediately we have $r_{j+2} < r_{j+1} \le \frac{1}{2}r_j$. So suppose that $r_{j+1} > \frac{1}{2}r_j$. In that case the next division gives: $r_j = 1 \cdot r_{j+1} + r_{j+2}$, and so $r_{j+2} = r_j - r_{j+1} < \frac{1}{2}r_j$, as claimed.

We now return to the proof of the time estimate. Since every two steps must result in cutting the size of the remainder at least in half, and since the remainder never gets below 1, it follows that there are at most $2 \cdot [log_2 a]$ divisions. This is $O(log\, a)$. Each division involves numbers no larger than a, and so takes $O(log^2 a)$ bit operations. Thus, the total time required is $O(log\, a) \cdot O(log^2 a) = O(log^3 a)$. This concludes the proof of the proposition.

Remark. If one makes a more careful analysis of the number of bit operations, taking into account the decreasing size of the numbers in the successive divisions, one can improve the time estimate for the Euclidean algorithm to $O(log^2 a)$.

Proposition I.2.2. *Let $d = g.c.d.(a, b)$, where $a > b$. Then there exist integers u and v such that $d = ua + bv$. In other words, the g.c.d. of two numbers can be expressed as a linear combination of the numbers with integer coefficients. In addition, finding the integers u and v can be done in $O(log^3 a)$ bit operations.*

Outline of proof. The procedure is to use the sequence of equalities in the Euclidean algorithm from the bottom up, at each stage writing d in terms of earlier and earlier remainders, until finally you get to a and b. At each stage you need a multiplication and an addition or subtraction. So it is easy to see that the number of bit operations is once again $O(log^3 a)$.

Example 1 (continued). To express 7 as a linear combination of 1547 and 560, we successively compute:

$$\begin{aligned}
7 &= 28 - 1 \cdot 21 = 28 - 1(133 - 4 \cdot 28) \\
&= 5 \cdot 28 - 1 \cdot 133 = 5(427 - 3 \cdot 133) - 1 \cdot 133 \\
&= 5 \cdot 427 - 16 \cdot 133 = 5 \cdot 427 - 16(560 - 1 \cdot 427) \\
&= 21 \cdot 427 - 16 \cdot 560 = 21(1547 - 2 \cdot 560) - 16 \cdot 560 \\
&= 21 \cdot 1547 - 58 \cdot 560.
\end{aligned}$$

Definition. We say that two integers a and b are *relatively prime* (or that "a is prime to b") if $g.c.d.(a, b) = 1$, i.e., if they have no common