A Lower Bound for Integer Greatest Common Divisor Computations

YISHAY MANSOUR

Massachusetts Institute of Technology, Cambridge, Massachusetts

AND

BARUCH SCHIEBER AND PRASOON TIWARI

T.J. Watson Research Center, Yorktown Heights, New York

Abstract. It is proved that no finite computation tree with operations $\{+, -, *, /, \mathrm{mod}, <\}$ can decide whether the greatest common divisor (gcd) of a and b is one, for all pairs of integers a and b. This settles a problem posed by Grötschel et al. [6]. Moreover, if the constants explicitly involved in any operation performed in the tree are restricted to be "0" and "1" (and any other constant must be computed), then we prove an $\Omega(\log \log n)$ lower bound on the depth of any computation tree with operations $\{+, -, *, /, \mathrm{mod}, <\}$ that decides whether the gcd of a and b is one, for all pairs of n-bit integers a and b.

A novel technique for handling the truncation operation is implicit in the proof of this lower bound. In a companion paper [11], other lower bounds for a large class of problems are proved using a similar technique.

Categories and Subject Descriptors: F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems—*number-theoretic computations*;

General Terms: Algorithms

Additional Key Words and Phrases: Floor operation, greatest common divisor, lower bound, mod operation, truncation.

1. *Introduction*

The problem of computing the greatest common divisor (gcd) of two integers is one of the oldest computational problems. The first known algorithm to solve this problem was the Euclidean algorithm, which was discovered more than two

A preliminary version of this paper appears in the *Proceedings of the 29th Annual Symposium on Foundations of Computer Science* (White Plains, New York). IEEE, New York, 1988.

Y. Mansour was supported by an ISEF fellowship and National Science Foundation (NSF) contract CCR 86-57527. Part of the research of this author was done while visiting IBM's T J. Watson Research Center.

Author's addresses: Y. Mansour, Aiken Computation Lab, Harvard University, Cambridge, MA 02138; B. Schieber, IBM Research Division, T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598; P. Tiwari, Department of Computer Science, University of Wisconsin, 1210 West Dayton St., Madison, WI 53706.

thousand years ago. Interestingly, this algorithm has survived to the present day. Knuth [9] calls it "the granddaddy of all algorithms."

The question whether this algorithm and its variants are the best possible is an important open problem. Much effort has been devoted to analyzing the Euclidean algorithm and to binding its behavior. For an excellent summary, we refer the reader to [9]. We mention only two results: the first due to G. Lamé (1845), who proved that the worst-case running time of the Euclidean algorithm with division steps is linear in the length of the input, and the second due to Collins [4], who analyzed the average behavior of the same algorithm.

Instead of studying the behavior of an *algorithm*, we provide a nontrivial lower bound on the complexity of the gcd *problem* on a model of computation. Our lower bound applies to the problem of deciding if a pair of integers a and b, is relatively prime, that is, is $\gcd(a, b) = 1$? Note that the problem of computing the gcd is at least as hard as this relative primality testing problem.

The model of computation considered in this paper is the computation tree [2, 14, 20]. We prove that no finite computation tree with operations $\{+, -, *, /, \mathsf{mod}\}$ can decide whether a and b are relatively prime, for all pairs of integers a and b. Moreover, if the constants explicitly involved in any operation performed in the tree are restricted to "0" and "1" (and any other constant must be computed), then we prove an $\Omega(\log\log n^{1})$ lower bound on the depth of any computation tree with operations $\{+, -, *, /, \mathsf{mod}\}$, that decides whether a and b are relatively prime, for all pairs of n-bit integers $a > b > 0$. A novel technique for handling the truncation operation is implicit in the proof of the lower bound mentioned above.

Remark. Throughout the paper "/" stands for exact division, for example, $8/5 = 1.6$. Observe that division with truncation ("integer division") can be implemented using "/" and "mod" in a constant number of steps. In addition, all sets of operations in this paper are assumed to include the "$<$" comparison operation, which is never explicitly specified.

Grötschel et al. [6], while working on their book *Geometric Algorithms and Combinatorial Optimization*, asked if there exists a strongly polynomial algorithm for the gcd problem. (See [6, pp. 32–33, p. 225].) Notice that since there are only two integer inputs to the gcd problem, any strongly polynomial algorithm for this problem must have a constant number of arithmetic operations. A result of Stockmeyer [18] implies that there is no strongly polynomial algorithm for the gcd problem when the set of operations is restricted to $\{+, -, *, /\}$. The results of this paper imply that there is no strongly polynomial algorithm for the gcd problem even when the set of operations is extended to $\{+, -, *, /, \mathsf{mod}\}$.

In the rest of the introduction, we briefly review some of the previous work relevant to this paper. The computation tree is a nonuniform model of computation, while the Random Access Machine (RAM) is the corresponding uniform model of computation, and hence, weaker in this sense. On the other hand, a RAM is capable of indirect addressing while a computation tree is not. However, the power of indirect addressing has not been characterized, and it is not known whether it is a substantial advantage. The power of RAM has been extensively studied [15, 17]. A RAM of time complexity T with operations

[1] The base of all logarithms in the paper is two.

from a set (of arithmetic and boolean operations) OP, and *without* indirect addressing, can be simulated by a computation tree of depth $O(T)$ with operations from the set OP. (See, e.g., [13].) This implies that our lower bound holds for RAMs without indirect addressing. In a companion paper [11], we show that a similar lower-bound holds for RAMs with indirect addressing.

To the best of our knowledge, prior to this work, no nontrivial lower bounds were known on the depth of computation trees with (i) operations from the set $\{+, -, *, /, \mathsf{mod}\}$, and (ii) a constant number of integer inputs. Perhaps, one reason for the lack of progress in this area is that this set of operations does not possess nice algebraic properties. It is worth pointing out that any decision problem with an n-bit input can be solved by a computation tree of depth $O(n)$, that uses only the comparison operation. In addition, since the length of numbers is not restricted in the computation, even the weaker uniform model, that is, RAM without indirect addressing, can simulate any PSPACE-computation in polynomial time [3, 14, 17]. Moreover, it is known that hard problems, for example, factoring, can be solved on such a RAM, in *linear* time [16].

For the model of *algebraic* computation trees (i.e., computation trees that use only *algebraic* operations, including $+$, $-$, $*$, and $/$), good lower bound techniques are known when all the inputs are either real or rational numbers [2, 21]. On the other hand, few results are known when the inputs are restricted to be integers (as in our case). Paul and Simon prove an $\Omega(n \log n)$ lower bound for sorting n integers [13]. (See also [5].) Their result also applies to RAMs with indirect addressing. They also show that the sorting can be done in $O(n)$ steps if boolean operations are allowed.

Stockmeyer [18] proves a $\Theta(n)$ bound on the depth of any algebraic computation tree with the operations $\{+, -, *, /\}$, that decides if a given n-bit integer is odd. This implies a $\Theta(n)$ bound on the depth of any algebraic computation tree that decides if (i) a divides b, or (ii) $\gcd(a, b) = 1$, and on the depth of any algebraic *computation* tree [2, 19, 20] that computes (i) $\gcd(a, b)$, (ii) $a \bmod b$, (iii) $\lfloor a/b \rfloor$, or (iv) bitwise $\circ$ of a and b, for $\circ \in \{\mathsf{and}, \mathsf{or}, \mathsf{exclusive\text{-}or}\}$; for all n-bit integers a and b.

The nature of the mod operation, which is the main object of study in this paper, is examined in [1], [7] and [8]. Moran et al. [12] prove tight bounds on the depth of decision trees with arbitrary queries (including queries involving the mod operation). However, these bounds are nontrivial only when the number of inputs is nonconstant.

The paper is organized as follows: Section 2 includes some preliminary definitions. Sections 3 and 4 are devoted to the proof of the $\Omega(\log \log n)$ lower bound on the depth of any computation tree with operations $\{+, -, *, /, \mathsf{mod}\}$, that decides if a and b are relatively prime for all pairs of n-bit integers $a > b > 0$. The proof is presented in two parts. First, in Section 3, we prove a nonconstant lower bound on the depth of any computation tree that tests relative primality, that is, no fixed, finite depth computation tree can solve the relative primality problem for all pairs of integers. Then, the $\Omega(\log \log n)$ bound is obtained in Section 4 by a careful analysis of the proof in Section 3. Finally, in Section 5 we summarize our results and list some open problems.

2. *Preliminaries*

In this section, we first recall the definition of the computation tree model. Then, we define some properties of polynomials and rational expressions.

Finally, we define a lexicographic order on the set of bivariate polynomials and prove some of its properties.

2.1. The Computation Tree Model. We assume that the reader is familiar with the computation tree model. (See, e.g., [2] and [20].) Below, we briefly recall this model, and define some additional terminology used throughout the paper in relation to the model.

A computation tree T for a two input problem is a tree with labeled vertices. The label of vertex v is denoted f_v. The tree T has four types of vertices:

(1) *Input vertices.* The input vertices are the vertices in the first two levels of T. Each of these two levels has exactly one vertex. The two input vertices are labeled by the input variables (i.e., for input a, b, we have $f_r = a$ and $f_s = b$, where r is the root and s is its only child).
(2) *Computation vertices.* Each computation vertex v is labeled with a binary operation $f_v = g \circ h$, and has only one child. Throughout this paper, $g, h \in \mathcal{C} \cup \{f_\mu \mid \mu$ is an ancestor of v in $T\}$ and $\circ \in OP$, where $\mathcal{C}$ is the set of available constants, and OP is the set of available operations. In this paper, $OP = \{+, -, *, /, \mathrm{mod}\}$. The set $\mathcal{C}$ will be restricted to either the set of rationals (2), or the set $\{0, 1\}$.
(3) *Comparison vertices.* Each comparison vertex v is labeled with $g : h$, where, again, $g, h \in \mathcal{C} \cup \{f_\mu \mid \mu$ is an ancestor of v in $T\}$. Each comparison vertex has two children.
(4) *Output vertices.* The output vertices are the leaves of T. Each leaf v of T is labeled with an element from the set $\{0, 1\}$.

The computation for input (a, b) starts at the root of the tree T. When it arrives at a computation vertex v, the function $f_v = g \circ h$ is evaluated at the input (a, b) by computing $g \circ h$, and then the computation proceeds to the only child of v. When the computation arrives at a comparison vertex labeled with $g : h$, the functions g and h are evaluated. The computation proceeds to the left child if $g < h$ at the point (a, b), and to the right child, otherwise. The computation terminates at a leaf by producing the value of the label associated with it as the output.

A computation tree is said to solve a decision problem, if it produces the correct answer for each instance of the problem.

We remark that when $OP = \{+, -, *, /\}$, we can associate a rational expression $r_v(x, y) \in \mathcal{Q}(x, y)$, to each vertex v, which is either a computation vertex or a leaf. This label will have the following interpretation: *For all inputs* (a, b), *if the computation on the tree* T *with input* (a, b) *arrives at the vertex* v, *then* $f_v = r_v(a, b)$.

2.2. Polynomials and Rational Expressions. In the sequel, we consider the degree and the height of polynomials and rational expressions.

The *degree* of a polynomial $P(x, y)$ with respect to a variable x, denoted $\deg_x(P)$, is the maximum exponent of x appearing in any monomial of $P(x, y)$.

The *degree* of P, denoted $\deg(P)$, is $\max\{\deg_x(P), \deg_y(P)\}$. The *height* of P, denoted $\mathrm{hgt}(P)$, is the maximum among the absolute values of the coefficients of P.

For a set Λ of polynomials, the *degree* of Λ, denoted $\deg(\Lambda)$, is defined by $\deg(\Lambda) = \max_{P\in\Lambda}\deg(P)$. Similarly, the *height* of Λ, denoted $\mathrm{hgt}(\Lambda)$, is $\mathrm{hgt}(\Lambda) = \max_{P\in\Lambda}\mathrm{hgt}(P)$.

For a rational expression $R(x, y) = P(x, y)/Q(x, y)$, where P and Q are bivariate polynomials, define the *degree* of R to be the larger of the degrees of P and Q. Similarly, define the *height* of R to be the larger of the heights of P and Q. Notice that in our definition of the degree and the height of rational expressions cancellations are not allowed. For example, the degree of $R(x, y) = P(x, y)/P(x, y)$ will be the degree of P and not zero, and the height of R will be the height of P and not one.

LEMMA 2.1. *Let $P(x, y)$ and $Q(x, y)$ be two bivariate polynomials. Then, for $\circ \in \{+, -\}$, $deg(P\circ Q) \le \max\{deg(P), deg(Q)\}$ and $hgt(P\circ Q) \le hgt(P) + hgt(Q)$; $deg(P\circ Q) \le deg(P) + deg(Q)$ and $hgt(P\circ Q) \le (1 + deg(P))(1 + deg(Q))hgt(P)hgt(Q)$.*

PROOF. We only prove the bound on $\mathrm{hgt}(P \cdot Q)$. Observe that when we multiply two bivariate polynomials, each of the coefficients of the product is the sum of at most $(1 + \deg(P))(1 + \deg(Q))$ terms, each of them being the product of a coefficient in P by a coefficient in Q. The bound on $\mathrm{hgt}(P\circ Q)$ follows.

The other bounds are also straightforward. □

LEMMA 2.2. *Let $R(x, y) = P_1(x, y)/P_2(x, y)$ and $S(x, y) = Q_1(x, y)/Q_2(x, y)$ be two bivariate rational expressions. Then, for $\circ \in \{+, -\}$, $deg(R\circ S) \le deg(R) + deg(S)$ and $hgt(R\circ S) \le 2(1 + deg(R))\ (1 + deg(S))hgt(R)hgt(S)$; $deg(R\circ S) \le deg(R) + deg(S)$ and $hgt(R\circ S) \le (1 + deg(R))(1 + deg(S))hgt(R)hgt(S)$.*

PROOF. The bounds follow from Lemma 2.1 and from the fact that for $\circ \in \{+, -\}$, $R\circ S = P_1Q_2\circ Q_1P_2/P_2Q_2$. □

2.3. A LEXICOGRAPHIC ORDER ON THE BIVARIATE POLYNOMIALS. We define a lexicographic order on the set of the bivariate polynomials. For this purpose, we use the following lexicographic order on the set of bivariate monomials.

Definition 1. For two monomials cx^iy^j and dx^ky^l, $cx^iy^j > dx^ky^l$ if either (1) $i > k$ or (2) $i = k$ and $j > l$ or (3) $i = k$, $j = l$ and $|c| > |d|$.

We say that a polynomial is written in its *normal form* if it is written as a minimal sum of monomials, and these monomials are sorted in descending lexicographic order. Throughout this paper, we assume that all polynomials are written in their normal form.

Definition 2. For two bivariate polynomials $P(x, y)$ and $Q(x, y)$, $P(x, y) > Q(x, y)$ if, when written in their normal forms, there exists some $i \ge 1$, such that (the ith monomial in P) $>$ (the ith monomial in Q), and all the monomials preceding it are identical in both P and Q.

Given a polynomial $P(x, y)$, let the *leading monomial* of $P(x, y)$ be the first monomial in the normal form of $P(x, y)$. Let the *leading coefficient* of $P(x, y)$ be the coefficient of this monomial.

Below, we relate the lexicographic order defined on the polynomials, and the order among their values at certain points.

LEMMA 2.3. *For each bivariate polynomial $P(x, y)$, there exist positive integers $\pi_1(P)$ and $\pi_2(P)$ such that for all (a, b) satisfying $a > b^{\pi_1(P)}$ and $b > \pi_2(P)$, the sign of $P(a, b)$ is the same as the sign of the leading coefficient of $P(x, y)$. Furthermore, $\pi_1(P) \leq deg_y(P) + 1$, and $\pi_2(P) \leq 2M/L$, where L is the leading coefficient of P and M is the height of P.*

PROOF. Let $P(x, y) = \sum_{k=0}^{m} L_k x^{i_k} y^{j_k}$, where $P(x, y)$ is written in its normal form, and L_k is the coefficient of the kth monomial ($L_0 = L$). Denote $t_k(x, y) = x^{i_k} y^{j_k}$. Let $\pi_1(P) = 1 + \max_{0 \leq k \leq m-1}\{j_{k+1} - j_k\}$. Clearly, $\pi_1(P) \leq \deg_y(P) + 1$. From the lexicographic order it follows that $t_{k+1}(x, y)/t_k(x, y) = x^i y^j$, where either $i < 0$ *and* $j < \pi_1(P)$ or $i = 0$ *and* $j < 0$. Thus, for all positive (a, b), if $a > b^{\pi_1(P)}$ then $t_{k+1}(x, y)/t_k(x, y) < 1/b$, and hence, $t_k(x, y)/t_0(x, y) < 1/b^k$. Suppose that $L > 0$, we show that for all (a, b) satisfying $a > b^{\pi_1(P)}$ and $b > \pi_2(P)$, $P(a, b)$ is also positive. For all positive (a, b), such that $a > b^{\pi_1(P)}$,

$$P(a, b) \geq Lt_0(a, b) - \sum_{k=1}^{m} |L_k| x^{i_k} y^{j_k}$$
$$> Lt_0(a, b)\left(1 - \sum_{k=1}^{m} \frac{|L_k|}{Lb^k}\right).$$

To complete the proof, we show that for $b > 2M/L$, $(1 - \sum_{k=1}^{m} |L_k|/Lb^k) > 0$ or, equivalently, $\sum_{k=1}^{m} |L_k|/Lb^k < 1$. This latter inequality follows from

$$\sum_{k=1}^{m} \frac{|L_k|}{Lb^k} < \frac{M}{L} \sum_{k=1}^{m} b^{-k}$$
$$< \frac{M}{L} \sum_{k=1}^{m} \left(\frac{L}{2M}\right)^k < \frac{2M}{2(2M - L)} < 1.$$

Similarly, we can prove the Lemma for the case $L < 0$. □

From Lemma 2.3, it follows that for any two polynomials $Q(x, y)$ and $R(x, y)$ there exist positive integers $\pi_1(Q - R)$ and $\pi_2(Q - R)$ such that for all (a, b) satisfying $a > b^{\pi_1(Q-R)}$ and $b > \pi_2(Q - R)$ either always $Q(a, b) < R(a, b)$ or always $Q(a, b) \geq R(a, b)$.

3. *The $\omega(1)$ Lower Bound for an Arbitrary Set of Constants*

In this section, we prove a nonconstant lower bound on the depth of any computation tree with $OP = \{+, -, *, /, \mathrm{mod}\}$, that computes the gcd of all pairs of n-bit integers. We assume that all constants explicitly involved in any operation performed in the tree are rational numbers. In the next section, we restrict the constants explicitly involved in any operation to be "0" and "1", and prove an $\Omega(\log \log n)$ lower bound by a careful analysis of the proof given below.

THEOREM 3.1. *There is no (finite depth) computation tree with $OP = \{+, -, *, /, \mathrm{mod}\}$, that decides if a and b are relatively prime, for all pairs of integers $a > b > 0$.*

We begin with some notation and definitions.

Definition 3. Let r, α_1, α_2, and α_3 be nonnegative integers. Let $\Delta = (\delta_1, \delta_2, \ldots, \delta_r)$ and $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_r)$ be r-dimensional vectors of positive integers. For positive integers u_0, u_1, u_r, and u_{r+1}, the pair (u_0, u_1) is $\langle r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda\rangle$-*generated* by the pair (u_r, u_{r+1}) if there exist positive integers $u_2, u_3, \ldots, u_{r-1}$ such that:

$$
\begin{aligned}
u_0 &= \lambda_1 (u_1)^{\delta_1} + u_2, && (1)\\
u_1 &= \lambda_2 (u_2)^{\delta_2} + u_3,\\
&\vdots\\
u_i &= \lambda_{i+1} (u_{i+1})^{\delta_{i+1}} + u_{i+2},\\
&\vdots\\
u_{r-1} &= \lambda_r (u_r)^{\delta_r} + u_{r+1},\\
u_r &> (u_{r+1})^{\alpha_1}, && (2)\\
u_{r+1} &> \alpha_2,\\
u_r &\equiv u_{r+1} \equiv 1 \pmod{\alpha_3}. && (3)
\end{aligned}
$$

In this case, (u_r, u_{r+1}) is the $\langle r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda\rangle$-*generator* of (u_0, u_1), and $u_2, u_3, \ldots, u_{r+1}$ is the $\langle r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda\rangle$-*generating sequence* for (u_0, u_1).

Notice that eqs. (1) and (2) imply that $u_0 > u_1 > u_2 > u_3 > \cdots > u_r > u_{r+1}$.

Definition 4. Let $S(r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda)$ denote the following set of ordered pairs of positive integers:

$$
\begin{aligned}
\{(u_0, u_1) &: \text{there exist integers } u_r, u_{r+1} \text{ such that } (u_0, u_1)\\
&\text{is } \langle r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda\rangle\text{-generated by } (u_r, u_{r+1})\}. \quad (4)
\end{aligned}
$$

For convenience, we omit the null vectors Δ and Λ whenever $r = 0$. In this case, the set $S(0, \alpha_1, \alpha_2, \alpha_3)$ consists of all pairs (u_0, u_1) such that $u_0 > (u_1)^{\alpha_1}$, $u_1 > \alpha_2$, $u_0 \equiv u_1 \equiv 1 \pmod{\alpha_3}$ (in accordance with the above definition).

Perhaps the most important characteristic of the above definitions is its similarity to the Euclidean algorithm for solving the gcd problem. As an immediate consequence of the definition, we get the two properties stated below. These properties are the key to our proof strategy.

LEMMA 3.2. THE CORRESPONDENCE PROPERTY. *There is an one-to-one correspondence between the elements of the sets* $S(r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda)$ *and* $S(0, \alpha_1, \alpha_2, \alpha_3)$. *Specifically, each pair* $(u_0, u_1) \in S(r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda)$ *corresponds to the unique pair* $(u_r, u_{r+1}) \in S(0, \alpha_1, \alpha_2, \alpha_3)$ *such that* (u_r, u_{r+1}) *is the* $\langle r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda\rangle$-*generator of* (u_0, u_1). *Furthermore, if* (u_0, u_1) *corresponds to* (u_r, u_{r+1}), *then* $gcd(u_0, u_1) = gcd(u_r, u_{r+1})$.

PROOF. Let $a_2 > a_3 > \cdots > a_{r+1}$ and $b_2 > b_3 > \cdots > b_{r+1}$, be the generating sequences for (a_0, a_1) and (b_0, b_1), respectively. It is easy to check that if $(a_i, a_{i+1}) \neq (b_i, b_{i+1})$ for some $0 \leq i \leq r$, then $(a_j, a_{j+1}) \neq (b_j, b_{j+1})$ for any j, $0 \leq j \leq r$. The assertion about the gcd's follows from the Euclidean algorithm. □

LEMMA 3.3. THE CONTAINMENT PROPERTY. *Let* Δ' *and* Λ' *be* $(r+1)$-*dimensional vectors of positive integers, obtained from r-dimensional*

vectors Δ *and* Λ *by appending positive integers* δ *and* λ, *respectively. Then,* $S(r+1, 1, \alpha_2, \alpha_3, \Delta', \Lambda') \subseteq S(r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda)$, *provided* $\delta \geq \alpha_1$ *and* $\lambda \equiv 0 \pmod{\alpha_3}$.

PROOF. Suppose that $(a_0, a_1) \in S(r+1, 1, \alpha_2, \alpha_3, \Delta', \Lambda')$, and let $a_2 > a_3 > \cdots > a_r > a_{r+1} > a_{r+2}$ be its generating sequence. By definition, $a_r = \lambda a_{r+1}^{\delta} + a_{r+2}$. Therefore, $a_r \equiv 1 \pmod{\alpha_3}$, and $a_r > a_{r+1}^{\alpha_1}$. In addition, $a_{r+1} > a_{r+2}^{\alpha_1} > \alpha_2$. Hence, $(a_0, a_1) \in S(r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda)$. □

LEMMA 3.4. $S(0, \alpha_1, \alpha_2, \alpha_3)$ *contains two pairs* (a_0, a_1) *and* (b_0, b_1), *such that* $\gcd(a_0, a_1) \neq 1$ *and* $\gcd(b_0, b_1) = 1$.

PROOF. Let e be the least exponent such that $(1+\alpha_3)^e > \alpha_2$. Define $(a_0, a_1) = ((1+\alpha_3)^{(e+1)\alpha_1}, (1+\alpha_3)^e)$, and $(b_0, b_1) = (1 + \alpha_3(1+\alpha_3)^{(e+1)\alpha_1}, (1+\alpha_3)^e)$. Clearly, $\gcd(a_0, a_1) = (1+\alpha_3)^e \neq 1$, and $\gcd(b_0, b_1) = 1$. □

The proof of Theorem 3.1 is based on the following lemma:

LEMMA 3.5. *Let* T *be a computation tree with the operations* $\{+, -, *, /, \mathrm{mod}\}$, *that decides if* a *and* b *are relatively prime, for all integers* $a > b > 0$. *Then, there is a path* $\mathcal{P}$ *from the root of* T *to a leaf, and a subset* $\mathcal{S}$ *of inputs, with the following properties*:

(1) $\mathcal{S} = S(r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda)$, *for some* $r, \alpha_1, \alpha_2, \alpha_3, \Delta$, *and* Λ; *and*
(2) *For each input* $(a, b) \in \mathcal{S}$, *the computation follows the path* $\mathcal{P}$.

Before proving Lemma 3.5, we show how it can be used to prove Theorem 3.1.

PROOF OF THEOREM 3.1. Suppose that we are given a computation tree T with the operations $\{+, -, *, /, \mathrm{mod}\}$, that decides if a and b are relatively prime, for all integers $a > b > 0$. By Lemma 3.5, there is a path $\mathcal{P}$ from the root of T to a leaf, and a set of inputs $\mathcal{S} = S(r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda)$, such that for each input $(a, b) \in \mathcal{S}$ the computation follows the the path $\mathcal{P}$. Let v be the leaf at the end of the path $\mathcal{P}$. By the definition of a computation tree, the output for each input whose computation terminates at the leaf v is the same as the label of v. But this contradicts Lemma 3.4, which asserts that the set $S(0, \alpha_1, \alpha_2, \alpha_3)$, and hence (by the Correspondence Property) the set $\mathcal{S}$, consists of some pairs that are relatively prime, and some that are not. Hence, the tree T does not decide the relative primality question for each pair of integer inputs. □

The most involved part of this section is the proof of Lemma 3.5 given below.

PROOF OF LEMMA 3.5. Let us denote the vertices on the path $\mathcal{P}$ by $v_1, v_2, \ldots, v_l$, in that order, where v_1 is the root of the tree T, v_i is a child of v_{i-1}, and v_l is a leaf of the tree T. We define the path $\mathcal{P}$ and the set $\mathcal{S}$ inductively, starting with the path v_1, v_2, v_3 (v_1 and v_2 are the input vertices and v_3 is the only child of v_2), and the set $\mathcal{S}^{(2)} = S(0, 1, 0, 1)$ (which consists of all pairs (a, b), where $a > b > 0$). As part of the induction hypothesis, we maintain some additional properties of the path and set under consideration. These properties are described below.

Suppose that we have (a) selected a prefix of $\mathscr{P}$, which starts at v_1, and ends at a vertex v_{i+1}, and (b) constructed the set $\mathscr{S}^{(i)} = S(r^{(i)}, \alpha_1^{(i)}, \alpha_2^{(i)}, \alpha_3^{(i)}, \Delta^{(i)}, \Lambda^{(i)})$, with the following properties:

(1) For each input $(a, b) \in \mathscr{S}^{(i)}$, the computation follows the path from the root to v_{i+1}.
(2) For each computation vertex v on the path from the root to the vertex v_{i+1}, excluding the vertex v_{i+1}, there is a pair of bivariate polynomials $(F_v^i(x, y), G_v^i(x, y))$ with integer coefficients, such that for each input $(a, b) \in \mathscr{S}^{(i)}$, $G_v^i(u, v) \neq 0$, and $f_v(a, b) = F_v^i(u, v)/G_v^i(u, v)$, where (u, v) is the $\langle r^{(i)}, \alpha_1^{(i)}, \alpha_2^{(i)}, \alpha_3^{(i)}, \Delta^{(i)}, \Lambda^{(i)}\rangle$-generator of (a, b).

We construct the set $\mathscr{S}^{(i+1)} \subseteq \mathscr{S}^{(i)}$ such that Property 2 is satisfied also for the vertex v_{i+1} and each input $(a, b) \in \mathscr{S}^{(i+1)}$. We also select an outgoing edge of v_{i+1} and prove that, for each input $(a, b) \in \mathscr{S}^{(i+1)}$, the computation follows this edge.

For the proof, we maintain two additional properties of the polynomials $(F_v^i(x, y), G_v^i(x, y))$ and the input set $\mathscr{S}^{(i)}$. These properties are:

(3) The leading coefficient of each of the polynomials $G_v^i(x, y)$ is positive.
(4) For each polynomial $F_v^i(x, y)$ and each $(a, b) \in \mathscr{S}^{(i)}$ the sign of $F_v^i(u, v)$ (where (u, v) is the $\langle r^{(i)}, \alpha_1^{(i)}, \alpha_2^{(i)}, \alpha_3^{(i)}, \Delta^{(i)}, \Lambda^{(i)}\rangle$-generator of (a, b)) is the same as the sign of the leading coefficient of $F_v^i(x, y)$. This also holds for all polynomials $G_v^i(x, y)$. (That is, $G_v^i(u, v) > 0$.)

By the definition of the tree T, either the value $f_{v_{i+1}} = g \circ h$ is computed, or the comparison $g : h$ is resolved, at the vertex v_{i+1}. Here, $g, h \in \mathscr{Q} \cup \{f_{v_j} \mid v_j$ is a computation vertex, $j \leq i\}$ and $\circ \in \{+, -, *, /, \mathrm{mod}\}$. We use the following notation in the rest of this proof:

$$\big(P_1(x, y), P_2(x, y)\big) = \begin{cases} (\text{numerator of } g, \text{denominator of } g) & \text{if } g \in \mathscr{Q}, \\ \big(F_v^i(x, y), G_v^i(x, y)\big) & \text{if } g = f_v, \end{cases}$$

and

$$\big(Q_1(x, y), Q_2(x, y)\big) = \begin{cases} (\text{numerator of } h, \text{denominator of } h) & \text{if } h \in \mathscr{Q} \\ \big(F_v^i(x, y), G_v^i(x, y)\big) & \text{if } h = f_v \end{cases}.$$

Finally, let

$$P(x, y) = P_1(x, y)Q_2(x, y),\ Q(x, y) = P_2(x, y)Q_1(x, y),$$

and

$$H(x, y) = P_2(x, y)Q_2(x, y).$$

The proof is based on a case-by-case analysis. In each case, we will define the next vertex v_{i+2} on the path $\mathscr{P}$, the parameters $r^{(i+1)}$, $\alpha_1^{(i+1)}$, $\alpha_2^{(i+1)}$, $\alpha_3^{(i+1)}$, $\Delta^{(i+1)}$, $\Lambda^{(i+1)}$, and the polynomials $(F_{v_j}^{i+1}(x, y), G_{v_j}^{i+1}(x, y))$, for each computation vertex $v_j \in \{v_1, v_2, \ldots, v_{i+1}\}$. Some of these parameters will be

defined explicitly, *while those not specified explicitly are assumed to take the value of the corresponding parameter with superscript* (i). After specifying these parameters, the set $\mathscr{S}^{(i+1)}$ will be defined to be $S(r^{(i+1)}, \alpha_1^{(i+1)}, \alpha_2^{(i+1)}, \alpha_3^{(i+1)}, \Delta^{(i+1)}, \Lambda^{(i+1)})$.

We note two things. First, our construction is such that the set $\mathscr{S}^{(i+1)}$ is a subset of $\mathscr{S}^{(i)}$. Second, the only case where $r^{(i+1)} \neq r^{(i)}$; that is, the dimensions of $\Delta^{(i+1)}$ and $\Lambda^{(i+1)}$ are not the same as the dimensions of $\Delta^{(i)}$ and $\Lambda^{(i)}$, is in case v_{i+1} is a mod vertex and some power of y appears in the leading monomial of $Q(x, y)$ (the last case of the proof).

Let us first resolve the case when v_{i+1} is a *comparison* vertex. Let $B(x, y) = P(x, y) - Q(x, y) \neq 0$. Then, Lemma 2.3 guarantees the existence of two positive integers $\pi_1(B)$ and $\pi_2(B)$, and $\bullet \in \{<, >\}$, such that for all pairs (a, b), where

$$a > b^{\pi_1(B)}, \qquad \text{and} \qquad b > \pi_2(B),$$
$$\frac{P_1(a, b)}{P_2(a, b)} \bullet \frac{Q_1(a, b)}{Q_2(a, b)}.$$

Thus, for all pairs (a, b) with this property, that arrive at the vertex v_{i+1}, the next vertex v_{i+2} is either left-child of v_{i+1} if $\bullet = <$, or the right-child of v_{i+1} if $\bullet = >$. Define $\alpha_1^{(i+1)} = \max(\alpha_1^{(i)}, \pi_1(B))$ and $\alpha_2^{(i+1)} = \max(\alpha_2^{(i)}, \pi_2(B))$. Let $\mathscr{S}^{(i+1)} = S(r^{(i+1)}, \alpha_1^{(i+1)}, \alpha_2^{(i+1)}, \alpha_3^{(i+1)}, \Delta^{(i+1)}, \Lambda^{(i+1)})$. Clearly, $\mathscr{S}^{(i+1)} \subseteq \mathscr{S}^{(i)}$, and for each $(a, b) \in \mathscr{S}^{(i+1)}$ and each $(F_{v_j}^{i+1}(x, y), G_{v_j}^{i+1}(x, y))$, Properties 1–4 are satisfied.

Next, consider the case when v_{i+1} is a computation vertex. In this case, the only child of v_{i+1} is chosen as v_{i+2}. However, the choice of the set $\mathscr{S}^{(i+1)}$ strongly depends on the particular operation $\circ$ performed at v_{i+1}.

Suppose that $\circ \in \{+, -\}$. This case is very similar to the case of a comparison vertex discussed above. Following the argument in that case, let $B(x, y) = P(x, y) \circ Q(x, y)$. Define $\alpha_1^{(i+1)} = \max(\alpha_1^{(i)}, \pi_1(B))$ and $\alpha_2^{(i+1)} = \max(\alpha_2^{(i)}, \pi_2(B))$. Let $\mathscr{S}^{(i+1)} = S(r^{(i+1)}, \alpha_1^{(i+1)}, \alpha_2^{(i+1)}, \alpha_3^{(i+1)}, \Delta^{(i+1)}, \Lambda^{(i+1)})$, and $F_{v+1}^{i+1}(x, y), G_{v_{i+1}}^{i+1}(x, y)) = (B(x, y), H(x, y))$. Clearly, $\mathscr{S}^{(i+1)} \subseteq \mathscr{S}^{(i)}$, and for each $(a, b) \in \mathscr{S}^{(i+1)}$ and each $(F_{v_j}^{i+1}(x, y), G_{v_j}^{i+1}(x, y))$, Properties 1–4 are satisfied.

Suppose that $\circ = *$. This is the simplest case. We just let $(F_{v_{i+1}}^{i+1}(x, y), G_{v_{i+1}}^{i+1}(x, y)) = (P_1(x, y)Q_1(x, y), H(x, y))$. Define $\mathscr{S}^{(i+1)} = \mathscr{S}^{(i)}$. Clearly, for each $(a, b) \in \mathscr{S}^{(i+1)}$ and each $(F_{v_j}^{i+1}(x, y), G_{v_j}^{i+1}(x, y))$, Properties 1–4 are satisfied.

Next, suppose that $\circ = /$. Let ρ be the *sign* of the leading coefficient of $Q_1(x, y)$. Define $(F_{v_{i+1}}^{i+1}(x, y), G_{v_{i+1}}^{i+1}(x, y)) = (\rho P(x, y), \rho Q(x, y))$, and $\mathscr{S}^{(i+1)} = \mathscr{S}^{(i)}$. This is the only case where $G_{v_{(i+1)}}^{i+1}(x, y)$ is not a product of $G_{v_j}^{i+1}(x, y)$'s, for $j \leq i$. Nevertheless, $G_{v_{i+1}}(u, v) \neq 0$ for any (u, v) that is the $\langle r^{(i+1)}, \alpha_1^{(i+1)}, \alpha_2^{(i)}, \alpha_3^{(i+1)}, \Delta^{(i+1)}, \Lambda^{(i+1)}\rangle$-generator of some $(a, b) \in \mathscr{S}^{(i+1)}$, because T is a well-defined computation tree that does not contain any division by zero. Clearly, for each $(a, b) \in \mathscr{S}^{(i+1)}$ and each $(F_{v_j}^{i+1}(x, y), G_{v_j}^{i+1}(x, y))$, Properties 1–4 are satisfied.

The only remaining case is when $\circ = \text{mod}$. The rest of this section is devoted to this case.

Before we continue, let us recall the definition of the mod operation. The mod operation is defined in the terms of the floor operation. For a real number r, $\lfloor r \rfloor$ is the greatest integer $\leq r$. For two *rational* numbers a and b, where $b \neq 0$, $a \bmod b = a - |b| \lfloor a/|b| \rfloor$. Note that $0 \leq (a \bmod b) < |b|$.

In the following discussion, we repeatedly use the fact that for polynomials $P_1(x, y)$, $P_2(x, y)$, $Q_1(x, y)$, and $Q_2(x, y)$, a rational function $R(x, y)$, and any two integers (u, v),

$$\frac{P_1(u, v)}{P_2(u, v)} \equiv R(u, v) \qquad \left(\bmod \frac{Q_1(u, v)}{Q_2(u, v)}\right),$$

if an only if $P(u, v) \equiv R(u, v)H(u, v) \pmod{Q(u, v)}$. (Recall that $P(x, y) = P_1(x, y)Q_2(x, y)$, $Q(x, y) = P_2(x, y)Q_1(x, y)$, and $H(x, y) = P_2(x, y)Q_2(x, y)$.)

Recall that for each $(a, b) \in \mathcal{S}^{(i)}$,

$$f_{v_{i+1}}(a, b) = \frac{P_1(u, v)}{P_2(u, v)} \bmod \frac{Q_1(u, v)}{Q_2(u, v)},$$

where (u, v) is the $\langle r^{(i)}, \alpha_1^{(i)}, \alpha_2^{(i)}, \alpha_3^{(i)}, \Delta^{(i)}, \Lambda^{(i)}\rangle$-generator of (a, b). In view of the above definition, we may assume, without loss of generality, that both the operands of any mod operation are nonnegative. Then, Properties 3 and 4 imply that the leading coefficients of $P(x, y)$ and $Q(x, y)$ are positive. We now show a way to construct $\mathcal{S}^{(i+1)} \subseteq \mathcal{S}^{(i)}$ such that for each input $(a, b) \in \mathcal{S}^{(i)}$ the value of $f_{v_{i+1}}$ is given by $F_{v_{i+1}}^{i+1}(\hat{u}, \hat{v})/G_{v_{i+1}}^{i+1}(\hat{u}, \hat{v})$, where the $(\hat{u}, \hat{v})$ is the $\langle r^{(i+1)}, \alpha_1^{(i+1)}, \alpha_2^{(i)}, \alpha_3^{(i+1)}, \Delta^{(i+1)}, \Lambda^{(i+1)}\rangle$-generator of (a, b) and $(F_{v_{i+1}}^{i+1}(x, y), G_{v_{i+1}}^{i+1}(x, y))$ is a pair of bivariate polynomials.

Let $d_x = \deg_x(Q)$, $d_y = \deg_y(Q)$, and let $Lx^{d_x}y^l$ be the leading monomial of $Q(x, y)$. We consider the following four cases in order: (1) $P(x, y) < Q(x, y)$; (2) $Q(x, y)$ is a constant, that is, $d_x = d_y = 0$; (3) No power of y appears in the leading monomial of $Q(x, y)$, that is, $l = 0$; and (4) $l > 0$.

Case 1. $P(x, y) < Q(x, y)$. Let $B(x, y) = Q(x, y) - P(x, y)$. Since $B(x, y) > 0$, Lemma 2.3 guarantees the existence of two positive integers $\pi_1(B)$, and $\pi_2(B)$ such that for each $(u, v) \in S(0, \pi_1(B), \pi_2(B), 1)$, $B(u, v) > 0$. Observe that $B(u, v) > 0$ implies that $Q(u, v) > P(u, v)$. Thus,

$$\frac{Q_1(u, v)}{Q_2(u, v)} > \frac{P_1(u, v)}{P_2(u, v)} \geq 0.$$

We conclude that for each $(u, v) \in S(0, \pi_1(B), \pi_2(B), 1)$,

$$\frac{P_1(u, v)}{P_2(u, v)} \bmod \frac{Q_1(u, v)}{Q_2(u, v)} = \frac{P_1(u, v)}{P_2(u, v)}.$$

Let $\alpha_1^{(i+1)} = \max(\alpha_1^{(i)}, \pi_1(B))$ and $\alpha_2^{(i+1)} = \max(\alpha_2^{(i)}, \pi_2(B))$. Define $\mathcal{S}^{(i+1)} = S(r^{(i+1)}, \alpha_1^{(i+1)}, \alpha_2^{(i+1)}, \alpha_3^{(i+1)}, \Delta^{(i+1)}, \Lambda^{(i+1)})$, and $(F_{v_{i+1}}^{i+1}(x, y), G_{v_{i+1}}^{i+1}(x, y)) = (P_1(x, y), P_2(x, y))$. Clearly, $\mathcal{S}^{(i+1)} \subseteq \mathcal{S}^{(i)}$, and for each $(a, b) \in \mathcal{S}^{(i+1)}$ and each $(F_{v_j}^{i+1}(x, y), G_{v_j}^{i+1}(x, y))$, Properties 1–4 are satisfied.

Case 2. $Q(x, y)$ is the constant (and, therefore, the positive integer) L. Recall that all coefficients of $P(x, y)$ are integers. Let $P(1, 1) = cL + \beta$,

where c is an integer and β is a nonnegative integer such that $0 \le \beta < L$. Let $\alpha_3^{(i+1)} = L\alpha_3^{(i)}$. Then, for each $(u, v) \in S(0, \alpha_1^{(i)}, \alpha_2^{(i)}, \alpha_3^{(i+1)})$, $P(u, v) = n_{uv}L + \beta$, where n_{uv} is an integer that depends on the pair (u, v). (Recall that $u \equiv v \equiv 1 \pmod{\alpha_3^{(i+1)}}$.) Dividing by $H(u, v)(= P_2(u, v)Q_2(u, v))$ we get

$$\frac{P_1(u, v)}{P_2(u, v)} = n_{uv}\frac{L}{H(u, v)} + \frac{\beta}{H(u, v)}.$$

Substituting for $L = Q(u, v)(= P_2(u, v)Q_1(u, v))$, we get

$$\frac{P_1(u, v)}{P_2(u, v)} = n_{uv}\frac{Q_1(u, v)}{Q_2(u, v)} + \frac{\beta}{H(u, v)}.$$

Since $0 \le \beta < L$, Property 3 implies that

$$0 \le \frac{\beta}{H(u, v)} < \frac{L}{H(u, v)} = \frac{Q_1(u, v)}{Q_2(u, v)}.$$

We conclude that for each $(u, v) \in S(0, \alpha_1^{(i)}, \alpha_2^{(i)}, \alpha_3^{(i+1)})$,

$$\frac{P_1(u, v)}{P_2(u, v)} \bmod \frac{Q_1(u, v)}{Q_2(u, v)} = \frac{\beta}{H(u, v)}.$$

Define $\mathcal{S}^{(i+1)} = S(r^{(i+1)}, \alpha_1^{(i+1)}, \alpha_2^{(i+1)}, \alpha_3^{(i+1)}, \Delta^{(i+1)}, \Lambda^{(i+1)})$, and $(F_{v_{i+1}}^{i+1}(x, y), G_{v_{i+1}}^{i+1}(x, y)) = (\beta, H(x, y))$. Clearly, $\mathcal{S}^{(i+1)} \subseteq \mathcal{S}^{(i)}$, and for each $(a, b) \in \mathcal{S}^{(i+1)}$ and each $(F_{v_j}^{i+1}(x, y), G_{v_j}^{i+1}(x, y))$, Properties 1–4 are satisfied.

Case 3. The leading monomial of $Q(x, y)$ is Lx^{d_x}, that is, no power of y appears in the leading monomial of $Q(x, y)$. Divide $P(x, y)$ by $Q(x, y)$ as polynomials in x. Corollary 4.5 of the following section implies that $P(x, y) = L^{-d}(A(x, y)Q(x, y) + R(x, y))$, where (i) $d = \deg_x(P) - \deg_x(Q) + 1$, (ii) all the coefficients of $A(x, y)$ and $R(x, y)$ are integers, and (iii) $\deg_x(R) < \deg_x(Q)$. Thus, $R \prec Q$.

As in Case 2, let $A(1, 1) = cL^d + \beta$, where c is an integer and β is a nonnegative integer such that $0 \le \beta < L^d$. Let $\alpha_3^{(i+1)} = L^d\alpha_3^{(i)}$. Then, for each $(u, v) \in S(0, \alpha_1^{(i)}, \alpha_2^{(i)}, \alpha_3^{(i)})$,

$$A(u, v) \equiv \beta \pmod{L^d}.$$

Hence, for each such pair (u, v),

$$P(u, v) \equiv L^{-d}\big(\beta Q(u, v) + R(u, v)\big) \pmod{Q(u, v)}.$$

Dividing by $H(u, v)(= P_2(u, v)Q_2(u, v))$, we get

$$\frac{P_1(u, v)}{P_2(u, v)} \equiv \frac{\beta Q(u, v) + R(u, v)}{L^d P_2(u, v)Q_2(u, v)} \left(\bmod \frac{Q_1(u, v)}{Q_2(u, v)}\right).$$

We distinguish between two subcases:

Subcase 3.1. $\beta > 0$. Consider the polynomial $L^{-d}(\beta Q(x, y) + R(x, y))$. Since $\deg_x(R) < \deg_x(Q)$ the leading coefficient of this polynomial is β/L^d times the leading coefficient of $Q(x, y)$ (which equals L). Let $B(x, y) = Q(x, y) - L^{-d}(\beta Q(x, y) + R(x, y))$. The leading coefficient of $B(x, y)$ is $(1 - \beta/L^d)L$. Since $0 < \beta/L^d < 1$, the leading coefficient of $B(x, y)$ is

positive. Using Lemma 2.3, let $\pi_1 = \max\{\pi_1(B), \pi_1(\beta Q + R)\}$, $\pi_2 = \max\{\pi_2(B), \pi_2(\beta Q + R)\}$. Then, $0 < L^{-d}(\beta Q(u, v) + R(u, v)) < Q(u, v)$, for all $u > v^{\pi_1}$, and $v > \pi_2$. This implies that

$$0 < \frac{\beta Q(u, v) + R(u, v)}{L^d P_2(u, v) Q_2(u, v)} < \frac{Q_1(u, v)}{Q_2(u, v)}.$$

We conclude that for each such (u, v),

$$\frac{P_1(u, v)}{P_2(u, v)} \bmod \frac{Q_1(u, v)}{Q_2(u, v)} = \frac{\beta Q(u, v) + R(u, v)}{L^d P_2(u, v) Q_2(u, v)}.$$

Let $\alpha_1^{(i+1)} = \max(\alpha_1^{(i)}, \pi_1)$, and $\alpha_2^{(i+1)} = \max(\alpha_2^{(i)}, \pi_2)$. Define $\mathscr{S}^{(i+1)} = S(r^{(i+1)}, \alpha_1^{(i+1)}, \alpha_2^{(i+1)}, \alpha_3^{(i+1)}, \Delta^{(i+1)}, \Lambda^{(i+1)})$, and $(F_{v_{i+1}}^{i+1}(x, y), G_{v_{i+1}}^{i+1}(x, y)) = (\beta Q(x, y) + R(x, y), L^d P_2(x, y) Q_2(x, y))$. Clearly, $\mathscr{S}^{(i+1)} \subseteq \mathscr{S}^{(i)}$, and for each $(a, b) \in \mathscr{S}^{(i+1)}$ and each $(F_{v_j}^{i+1}(x, y), G_{v_j}^{i+1}(x, y))$, Properties 1–4 are satisfied.

Subcase 3.2. $\beta = 0$. Define $\tilde{R}(x, y)$ as follows:

$$\tilde{R}(x, y) = \begin{cases} R(x, y) & \text{if the leading coefficient of } R(x, y) \geq 0, \\ L^d Q(x, y) + R(x, y) & \text{otherwise.} \end{cases}$$

The leading coefficients of the polynomials $\tilde{R}(x, y)$ and $L^d Q(x, y) - \tilde{R}(x, y)$ are positive. Using Lemma 2.3, let $\pi_1 = \max\{\pi_1(\tilde{R}), \pi_1(L^d Q - \tilde{R})\}$, $\pi_2 = \max\{\pi_2(\tilde{R}), \pi_2(L^d Q - \tilde{R})\}$. Then, $0 \leq \tilde{R}(u, v) < L^d Q(u, v)$, for all $u > v^{\pi_1}$, and $v > \pi_2$. This implies that

$$0 \leq \frac{\tilde{R}(u, v)}{L^d P_2(u, v) Q_2(u, v)} < \frac{Q_1(u, v)}{Q_2(u, v)}.$$

We conclude that for each such (u, v),

$$\frac{P_1(u, v)}{P_2(u, v)} \bmod \frac{Q_1(u, v)}{Q_2(u, v)} = \frac{\tilde{R}(u, v)}{L^d P_2(u, v) Q_2(u, v)}.$$

Let $\alpha_1^{(i+1)} = \max(\alpha_1^{(i)}, \pi_1)$, and $\alpha_2^{(i+1)} = \max(\alpha_2^{(i)}, \pi_2)$. Define $\mathscr{S}^{(i+1)} = S(r^{(i+1)}, \alpha_1^{(i+1)}, \alpha_2^{(i+1)}, \alpha_3^{(i+1)}, \Delta^{(i+1)}, \Lambda^{(i+1)})$, and $(F_{v_{i+1}}^{i+1}(x, y), G_{v_{i+1}}^{i+1}(x, y)) = (\tilde{R}(x, y), L^d P_2(x, y) Q_2(x, y))$. Clearly, $\mathscr{S}^{(i+1)} \subseteq \mathscr{S}^{(i)}$, and for each $(a, b) \in \mathscr{S}^{(i+1)}$ and each $(F_{v_j}^{i+1}(x, y), G_{v_j}^{i+1}(x, y))$, Properties 1–4 are satisfied.

Case 4. The leading monomial of $Q(x, y)$ is $Lx^{d_1} y^{l}$. Our goal is to reduce this case to Case 3 where no powers of y appear in the leading monomial. We introduce a new indeterminate z and substitute x using it. We substitute x by $\alpha_3^{(i)} y^{\alpha_1^{(i)}} + z$. Consider the polynomial $\hat{Q}(y, z) = Q(\alpha_3^{(i)} y^{\alpha_1^{(i)}} + z, y)$. Observe that the leading monomial in $\hat{Q}(y, z)$ is a constant times a power of y, that is, no power of z appears in the leading monomial of $\hat{Q}(y, z)$. Thus, we have reduced this case to Case 3 with only two differences: (a) instead of the set $\mathscr{S}^{(i)}$ of inputs there, we have the set $\hat{\mathscr{S}} = S(r^{(i+1)}, 1,$

$\alpha_2^{(i)}$, $\alpha_3^{(i)}$, $\Delta^{(i+1)}$, $\Lambda^{(i+1)}$), where $r^{(i+1)} = r^{(i)} + 1$; $\Delta^{(i+1)}$ and $\Lambda^{(i+1)}$ are the $(r+1)$-dimensional vectors obtained by appending $\alpha_1^{(i)}$ and $\alpha_3^{(i)}$ to $\Delta^{(i)}$ and $\Lambda^{(i)}$, respectively; (b) instead of the polynomials $(F_v^i(x, y), G_v^i(x, y))$, we have the polynomials $(F_v'(y, z), G_v'(y, z)) = (F_v^i(\alpha_3^{(i)} y^{\alpha_1^{(i)}} + z, y), G_v^i(\alpha_3^{(i)} y^{\alpha_1^{(i)}} + z, y))$.

By the Containment Property, $\hat{\mathscr{S}} \subseteq \mathscr{S}^{(i)}$. Hence, by an argument similar to that given in Case 3, we can define the parameters $\alpha_j^{(i+1)}$, for $j = 1, 2, 3$ and consequently the set $\mathscr{S}^{(i+1)} \subseteq \hat{\mathscr{S}}$, and the polynomials $(F_{v_j}^{i+1}(y, z), G_{v_j}^{i+1}(y, z))$, for $j = 1, \ldots, i+1$, that satisfy Properties 1–4. □

4. *The $\Omega(\log \log n)$ Lower Bound for the Constants $\{0, 1\}$*

In this section, we prove an $\Omega(\log \log n)$ lower bound on the depth of any computation tree with $OP = \{+, -, *, /, \mathrm{mod}\}$, that computes the gcd of all pairs of n-bit integers. We assume that "0" and "1" are the only constants explicitly involved in any operation performed in the tree (and that any other constant must be computed). In the following proofs, we refer to the proofs in the previous section.

In order to prove an $\Omega(\log \log n)$ lower bound, we need a modified version of Lemma 3.4 and upper bounds on the parameters in the proof of Lemma 3.5.

LEMMA 4.1. *Let $\alpha_1, \alpha_2, \alpha_3$, and t be positive integers such that $\alpha_1 < t$, and $\alpha_2, \alpha_3 < 2^t$. Then the set $S(0, \alpha_1, \alpha_2, \alpha_3) \cap \{(u, v) : 0 < u, v < 2^{2t(t+1)}\}$ contains two pairs (a_0, a_1) and (b_0, b_1), such that $\gcd(a_0, a_1) \neq 1$ and $\gcd(b_0, b_1) = 1$.*

PROOF. In order to prove this lemma, it is sufficient to prove each of the numbers a_0, a_1, b_0, and b_1 (constructed in the proof of Lemma 3.4), is less than $2^{2t(t+1)}$. If e is the least exponent such that $(1 + \alpha_3)^e > \alpha_2$, then $(1 + \alpha_3)^e < 2^{2t}$. The desired upper bounds are an immediate consequence of this observation. □

LEMMA 4.2. *Let T be a computation tree of depth h that decides if a and b are relatively prime, for all integers $2^n > a > b > 0$. Then, there is a path $\mathscr{P}$ from the root of T to a leaf, and a subset $\mathscr{S}$ of inputs, with the following properties:*

(1) *$\mathscr{S} = S(r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda) \cap \{(a, b) : 0 < a, b < 2^n\}$, for some positive integers r, α_1, α_2, α_3, $\delta_1, \delta_2, \ldots, \delta_r$, $\lambda_1, \lambda_2, \ldots, \lambda_r$, where $\Delta = (\delta_1, \delta_2, \ldots, \delta_r)$, and $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_r)$;*
(2) *For each input $(a, b) \in \mathscr{S}$, the computation follows the path $\mathscr{P}$;*
(3) *For each computation vertex v on the path $\mathscr{P}$, there is a pair of bivariate polynomials $(F_v(x, y), G_v(x, y))$ with integer coefficients, such that for each input $(a, b) \in \mathscr{S}$, $G_v(u, v) \neq 0$, and $f_v(a, b) = F_v(u, v)/G_v(u, v)$, where (u, v) is the $\langle r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda\rangle$-generator of (a, b); that is, the value computed at v on input $(a, b) \in \mathscr{S}$, is the value of the rational expression $F_v(x, y)/G_v(x, y)$ at (u, v); and*
(4) *Let $\Sigma = \{F_v(x, y), G_v(x, y) \mid v \in \mathscr{P}\}$. Define D and M to be the degree and the height of Σ, respectively. Then, $r \leq h$, $\max\{\alpha_1, D\} < 2^{2^{4h}}$, and $\max\{\alpha_2, \alpha_3, M\} < 2^{2^{2^{4h}}}$.*

Before proving Lemma 4.2, we show how it can be used to prove the main theorem of this section.

THEOREM 4.3. *Any computation tree with* $OP = \{+, -, *, /, \text{mod}\}$, *that decides if* a *and* b *are relatively prime, for all integers* $2^n > a > b > 0$, *must have depth* $\Omega(\log \log n)$.

PROOF. Suppose that we are given a computation tree T of depth $h < 1/4 \log\log(n^{1/5})$, with $OP = \{+, -, *, /, \text{mod}\}$, that decides if a and b are relatively prime, for all for all integers $2^n > a > b > 0$. By Lemma 4.2, we have the following: (i) there is a path $\mathscr{P}$ from the root of T to a leaf v, and a set of inputs $\mathscr{S} = S(r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda) \cap \{(a, b) : 0 < a, \ b < 2^n\}$, such that for each input $(a, b) \in \mathscr{S}$, the computation follows the path $\mathscr{P}$; (ii) each pair in $\mathscr{S}$ is relatively prime if the label $l \in \{0, 1\}$ of v is one; otherwise, none of the pairs in $\mathscr{S}$ is relatively prime; (iii) $\alpha_1 < 2^{2^{4h}} = n^{1/5}$, $\alpha_2, \alpha_3 < 2^{2^{2^{4h}}} = 2^{n^{1/5}}$, $\text{hgt}(F_v(x, y)/G_v(x, y)) < 2^{n^{1/5}}$, and $\deg(F_v(x, y)/G_v(x, y)) < n^{1/5}$.

Our goal is to arrive at a contradiction using Lemma 4.1.

Towards this end, let $t = n^{1/5} - 1$. We claim that each pair $(u, v) \in S(0, \alpha_1, \alpha_2, \alpha_3) \cap \{(u, v) : 1 \le u, v < 2^{2t(t+1)}\}$ generates a pair $(a, b) \in \mathscr{S} = S(r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda) = S(r, \alpha_1, \alpha_2, \alpha_3, \Delta, \Lambda) \cap \{(a, b) : 0 < u, \ v < 2^n\}$. Notice that the inputs a and b are "computed" by the first two vertices of the tree. If (u, v) is the generator of (a, b), then, by Lemma 4.2, a and b are polynomials in u and v of height and degree less than $2^{n^{1/5}}$ and $n^{1/5}$, respectively. This implies that the number of monomials in each of these polynomials is at most $n^{2/5}$ and that the value of each monomial is at most $2^{n^{1/5}}(2^{2n^{2/5}})^{2n^{1/5}}$. Therefore, $a, \ b \le n^{2/5}2^{n^{1/5}}2^{2n^{2/5}2n^{1/5}} = n^{2/5}2^{4n^{4/5}} < 2^n$, for large enough n.

But assertion (ii) in the first paragraph of this proof, together with the Correspondence Property, contradict Lemma 4.1 which asserts that some pairs in the set $(u, v) \in S(0, \alpha_1, \alpha_2, \alpha_3) \cap \{(u, v) : 1 \le u, \ v < 2^{2t(t+1)}\}$ are relatively prime, and some are not. □

PROOF OF LEMMA 4.2. As in the proof of Lemma 3.5, we denote the vertices on the path $\mathscr{P}$ by $v_1, v_2, \ldots, v_l$, in that order, where v_1 is the root of the tree T, v_i is a child of v_{i-1}, and v_l is a leaf of the tree T. In the proof of Lemma 3.5, the path $\mathscr{P}$ and the set $\mathscr{S}$ were defined inductively, starting with the path v_1, v_2, v_3 and the set $\mathscr{S}^{(2)} = S(0, 1, 0, 1)$ (which consists of all pairs (a, b), where $a > b > 0$). Following that proof, suppose that (a) we have selected a prefix of $\mathscr{P}$, which starts at v_1, and ends a vertex v_{i+1}, and (b) constructed the set $\mathscr{S}^{(i)} = S(r^{(i)}, \alpha_1^{(i)}, \alpha_2^{(i)}, \alpha_3^{(i)}, \Delta^{(i)}, \Lambda^{(i)})$ with the following properties:

(1) For each input $(a, b) \in \mathscr{S}^{(i)}$, the computation follows the path from the root to v_{i+1};
(2) For each computation vertex v on the path from the root to the vertex v_{i+1}, excluding the vertex v_{i+1}, there is a pair of bivariate polynomials $(F_v^i(x, y), G_v^i(x, y))$ with integer coefficients, such that for each input $(a, b) \in \mathscr{S}^{(i)}$, $G_v^i(u, v) \ne 0$, and $f_v(a, b) = F_v^i(u, v)/G_v^i(u, v)$, where (u, v) is the $\langle r^{(i)}, \alpha_1^{(i)}, \alpha_2^{(i)}, \alpha_3^{(i)}, \Delta^{(i)}, \Lambda^{(i)}\rangle$-generator of (a, b); and
(3) The leading coefficient of each of the polynomials $G_v^i(x, y)$ is positive.
(4) For each polynomial $F_v^i(x, y)$ and each $(a, b) \in \mathscr{S}^{(i)}$ the sign of $F_v^i(u, v)$ (where (u, v) is the $\langle r^{(i)}, \alpha_1^{(i)}, \alpha_2^{(i)}, \alpha_3^{(i)}, \Delta^{(i)}, \Lambda^{(i)}\rangle$-generator of (a, b)) is the same as the sign of the leading coefficient of $F_v^i(x, y)$. This also holds for all polynomials $G_v^i(x, y)$ (i.e., $G_v^i(x, v) > 0$).

(5) Let $\Sigma_i = \{F^i_{v_j}(x, y),\ G^i_{v_j}(x, y) \mid j \le i\}$. Define $D_i = \deg(\Sigma_i) + 2$ and $M_i = \mathrm{hgt}(\Sigma_i)$. Then, $r^{(i)} \le i$, $\max\{\alpha_1^{(i)},\ D_i\} < 2^{2^{4i}}$, and $\max\{\alpha_2^{(i)},\ \alpha_3^{(i)},\ M_i\} < 2^{2^{2^{4i}}}$.

As in the proof of Lemma 3.5, we construct the set $\mathcal{S}^{(i+1)} \subseteq \mathcal{S}^{(i)}$ such that Property 2 is satisfied also for the vertex v_{i+1} and each input $(a, b) \in \mathcal{S}^{(i+1)}$. We also select an outgoing edge of v_{i+1} and prove that for each input $(a, b) \in \mathcal{S}^{(i+1)}$ the computation follows this edge. It is easy to check that $r^{(i+1)} \le i + 1$. Therefore, in order to complete this proof, it is sufficient to show that the following two inequalities hold:

(i) $$\max\{\alpha_1^{(i+1)},\ D_{i+1}\} < 2^{2^{4(i+1)}},$$

(ii) $$\max\{\alpha_2^{(i+1)},\ \alpha_3^{(i+1)},\ M_{i+1}\} < 2^{2^{2^{4(i+1)}}}.$$

Now we follow the various cases in the proof of Lemma 3.5, and argue that Inequalities (i) and (ii) hold in each of them. Note that these inequalities hold for v_1 and v_2 because v_1 and v_2 are input vertices. (See the definition of a computation tree in Section 2.) In our arguments we repeatedly use Lemmas 2.1 and 2.2.

Let us first resolve the case when v_{i+1} is a *comparison* vertex. From the proof of Lemma 3.5, $D_{i+1} = D_i$, $M_{i+1} = M_i$, $\alpha_1^{(i+1)} = \max(\alpha_1^{(i)},\ \pi_1(P - Q))$, $\alpha_2^{(i+1)} = \max(\alpha_2^{(i)},\ \pi_2(P - Q))$, and $\alpha_3^{(i+1)} = \alpha_3^{(i)}$. Recall that each of P and Q is a product of two polynomials of degree less than D_i, and height at most M_i. Therefore, $\deg(P)$, $\deg(Q) < 2D_i$ and $\mathrm{hgt}(P)$, $\mathrm{hgt}(Q) \le D_i^2 M_i^2$. Now, Lemma 2.3 implies that $\pi_1(P - Q) \le 2D_i < 2^{2^{4(i+1)}}$, and $\pi_2(P - Q) \le 2M_i^2 D_i^2 < 2^{2^{2^{4(i+1)}}}$. Clearly, inequalities (i) and (ii) hold in this case.

Next, consider the case when v_{i+1} is a *computation* vertex. The following possibilities may arise:

Suppose that $\circ \in \{+, -\}$. The degree of $P \circ Q$ is less than $2D_i$, and its height is bounded by $2M_i^2 D_i^2$. From the proof of Lemma 3.5, it is clear that $D_{i+1} < 2D_i$, $M_{i+1} \le 2M_i^2 D_i^2$, $\alpha_1^{(i+1)} = \max(\alpha_1^{(i)},\ \pi_1(P \circ Q))$, $\alpha_2^{(i+1)} = \max(\alpha_2^{(i)},\ \pi_2(P \circ Q))$, and $\alpha_3^{(i+1)} = \alpha_3^{(i)}$. By an argument similar to the one given in the previous case, inequalities (i) and (ii) also hold in this case.

Suppose that $\circ = \{*, /\}$. From the proof of Lemma 3.5, it follows that: $D_{i+1} < 2D_i$, $M_{i+1} = M_i^2 D_i^2$, $\alpha_j^{(i+1)} = \alpha_j^{(i)}$, for $j = 1, 2, 3$. Therefore, inequalities (i) and (ii) hold in this case.

The only remaining case is when $\circ = \mathrm{mod}$. The rest of this section is devoted to this case.

Case 1. $P(x, y) < Q(x, y)$. From the proof of Lemma 3.5, it follows that $D_{i+1} = D_i$, $M_{i+1} = M_i$, $\alpha_1^{(i+1)} = \max(\alpha_1^{(i)},\ \pi_1(Q - P))$, $\alpha_2^{(i+1)} = \max(\alpha_2^{(i)},\ \pi_2(Q - P))$, and $\alpha_3^{(i+1)} = \alpha_3^{(i)}$. By an argument similar to that in the case of the comparison vertex, inequalities (i) and (ii) hold in this case.

Case 2. $Q(x, y)$ is the constant (and, therefore, the positive integer) L. Since L is the product of two constants of height at most M_i, $L \le M_i^2$. From the proof of Lemma 3.5 $\alpha_1^{(i+1)} = \alpha_1^{(i)}$, $\alpha_2^{(i+1)} = \alpha_2^{(i)}$, $D_{i+1} < 2D_i$, $M_{i+1} \le M_i^2 D_i^2$, and $\alpha_3^{(i+1)} = L\alpha_3^{(i)} \le M_i^2 \alpha_3^{(i)}$. Therefore, inequalities (i) and (ii) hold in this case.

Case 3. The leading monomial of $Q(x, y)$ is Lx^{d_x}, that is, no power of y appears in the leading monomial of $Q(x, y)$. In this case, we work with the bounds $\tilde{D}$ and $\tilde{M}$ instead of D_i and M_i, respectively. This is done in order to simplify the arguments in Case 4. For now, it is convenient to assume that $\tilde{D} = D_i$ and $\tilde{M} = M_i$.

Observe that $L \leq \tilde{D}\tilde{M}^2$, and $\deg_x(Q)(= d_x)$, $\deg_y(Q) \leq 2\tilde{D} - 2$. By Corollary 4.5, $\deg(R) < 4\tilde{D}^2$, and $\mathrm{hgt}(R) \leq (2\tilde{D})^{2\tilde{D}}(\tilde{D}^2\tilde{M}^2)^{2\tilde{D}}$.

From the proof of Lemma 3.5, it follows that $\alpha_3^{(i+1)} = L^{d_x}\alpha_3^{(i)} < (\tilde{D}\tilde{M}^2)^{2\tilde{D}}\alpha_3^{(i)}$. Therefore, $\alpha_3^{(i+1)}$ satisfies inequality (ii) in this case.

In Subcase 3.1, $\alpha_1^{(i+1)} = \max\{\alpha_1^{(i)}, \pi_1(B), \pi_1(\beta Q + R)\}$, $\alpha_2^{(i+1)} = \max\{\alpha_2^{(i)}, \pi_2(B), \pi_2(\beta Q + R)\}$, and two new polynomials, $\beta Q(x, y) + R(x, y)$ and $L^d P_2(x, y)Q_2(x, y)$ are added to the set Σ_{i+1}. The degree and height of each of these polynomials can be bounded as follows:

(1) $\deg(\beta Q + R) < 4\tilde{D}^2$, and $\mathrm{hgt}(\beta Q + R) \leq L^d\mathrm{hgt}(Q) + \mathrm{hgt}(R) < L^d\tilde{D}^2\tilde{M}^2 + (2\tilde{D})^{2\tilde{D}}(\tilde{D}^2\tilde{M}^2)^{2\tilde{D}}$.

(2) $\deg(L^d P_2 Q_2) < 2\tilde{D}$, and $\mathrm{hgt}(L^d P_2 Q_2) = L^d\mathrm{hgt}(P_2 Q_2) < L^d\tilde{D}^2\tilde{M}^2$.

(3) Recall that $L^d B(x, y) = L^d Q(x, y) - (\beta Q(x, y) - R(x, y))$. Therefore, $\deg(L^d B) < 4\tilde{D}^2$, and $\mathrm{hgt}(L^d B) \leq L^d\mathrm{hgt}(Q) + \mathrm{hgt}(R) < L^d\tilde{D}^2\tilde{M}^2 + (2\tilde{D})^{2\tilde{D}}(\tilde{D}^2\tilde{M}^2)^{2\tilde{D}}$. This implies that $\pi_1(B) = \pi_1(L^d B) \leq 4\tilde{D}^2$, and since $L^d B$ is a polynomial with integer coefficients also $\pi_2(B) \leq 2\mathrm{hgt}(L^d B)$.

We conclude that inequalities (i) and (ii) hold in this case.

In Subcase 3.2 $\alpha_1^{(i+1)} = \max(\alpha_1^{(i)}, \pi_1)$, $\alpha_2^{(i+1)} = \max(\alpha_2^{(i)}, \pi_2)$, where $\pi_1 = \max\{\pi_1(\tilde{R}), \pi_1(L^d Q - \tilde{R})\}$, $\pi_2 = \max\{\pi_2(\tilde{R}), \pi_2(L^d Q - \tilde{R})\}$, and two new polynomials, $\tilde{R}(x, y)$ and $L^d P_2(x, y)Q_2(x, y)$ are added to the set Σ_{i+1}. Notice that $\deg(\tilde{R})$, $\deg(L^d Q - \tilde{R}) < 4\tilde{D}^2$, and $\mathrm{hgt}(R)$, $\mathrm{hgt}(Q - R) < L^d\tilde{D}^2\tilde{M}^2 + (2\tilde{D})^{2\tilde{D}}(\tilde{D}^2\tilde{M}^2)^{2\tilde{D}}$. Therefore, by an argument similar to that in Subcase 3.1, inequalities (i) and (ii) hold in this case.

Case 4. The leading monomial of $Q(x, y)$ is $Lx^{d_x}y^l$. Recall that in order to reduce this case to Case 3, we substitute x by $\alpha_3^{(i)}y^\delta + z$ in all the polynomials of the set Σ_i, where $\delta \leq D_i$. Let Σ be the set of polynomials (in variables y and z) obtained by this substitution. Let $\tilde{D} = \deg(\tilde{\Sigma}) + 1$, and $\tilde{M} = \mathrm{hgt}(\tilde{\Sigma})$. It is easy to check that $\tilde{D} < D_i^2$ and $\tilde{M} < (\alpha_3^{(i)})^{D_i - 1}D_i M_i$. Then, the argument in Case 3 implies that the inequalities (i) and (ii) hold in this case. □

In Case 3 of the above proof and in the previous section, we asserted bounds on the degree and the height of the remainder polynomial $R(x, y)$. For completeness, we prove these well-known bounds in the following lemma.

LEMMA 4.4. *Let $P(x, y)$ and $Q(x, y)$ be two bivariate polynomials with integer coefficients bounded in absolute value by M and N, respectively. If Lx^d is the leading monomial of $Q(x, y)$, then $P(x, y) = 1/L^{\delta+1}A(x, y)Q(x, y) + 1/L^{\delta+1}R(x, y)$, where $A(x, y)$ and $R(x, y)$ are polynomials with integer coefficients, $hgt(R) \leq (2 + deg_y(Q))^{\delta+1}MN^{\delta+1}$, $\delta = \max\{-1, deg_x(P) - deg_x(Q)\}$, $deg_x(R) < deg_x(Q)\{= d\}$, $deg_y(R) \leq deg_y(P) + \delta deg_y(Q)$.*

PROOF. The proof is by induction on δ. The hypothesis holds for the basis case $\delta = -1$ with $A(x, y) = 0$ and $R(x, y) = P(x, y)$.

For the induction step, assume that the hypothesis holds for all $\delta < k$, for some $k > -1$. We prove it for k. Let $P(x, y) = p_1(y)x^e + p_2(y)x^{e-1} + \cdots$, be such that $k = e - d$. Consider the polynomial $S(x, y) = LP(x, y) - x^k p_1(y)Q(x, y)$. $\text{hgt}(S) \leq (2 + \deg_y(Q))MN$, $\deg_x(S) \leq \deg_x(P) - 1$, and $\deg_y(S) \leq \deg_y(P) + \deg_y(Q)$.

Applying the hypothesis to the pair $S(x, y)$ and $Q(x, y)$, yields $S(x, y) = (1/L^\delta)A(x, y)Q(x, y) + (1/L^\delta)R(x, y)$. Substituting for $S(x, y)$, we get $P(x, y) = 1/L^{\delta+1}(A(x, y) + L^\delta x^k p_1(y))Q(x, y) + 1/L^{\delta+1}R(x, y)$. In addition, $\text{hgt}(R) \leq (2 + \deg_y(Q))^\delta((2 + \deg_y(Q))MN)N^\delta = (2 + \deg_y(Q))^{\delta+1}MN^{\delta+1}$, $\deg_x(R) < \deg_x(Q)\{ = d\}$, and $\deg_y(R) \leq \deg_y(S) + (\delta - 1)\deg_y(Q) = \deg_y(P) + \delta \deg_y(Q)$. □

COROLLARY 4.5. *Let $P(x, y)$ and $Q(x, y)$ be two bivariate polynomials of degree less than an integer D, and integer coefficients bounded in absolute value by M. If Lx^d is the leading monomial of $Q(x, y)$, then $P(x, y) = 1/L^{\delta+1}A(x, y)Q(x, y) + 1/L^{\delta+1}R(x, y)$, where $A(x, y)$ and $R(x, y)$ are polynomials with integer coefficients,* $\text{hgt}(R) \leq (1 + D)^D M^{D+1}$, $\delta = \max\{-1, deg_x(P) - deg_x(Q)\}$, $deg_x(R) < deg_x(Q)\{ = d\}$, $deg_y(R) < D^2$.

5. *Conclusion*

We have proved an $\Omega(\log \log n)$ lower bound on the depth of any computation tree with operations from the set $\{+, -, *, /, \text{mod}\}$, that decides if a and b are relatively prime, for all pairs of n-bit integers a, b. We do not believe this bound to be tight, and a better lower bound for this problem would be interesting.

In a companion paper [11], we prove other lower bounds for a large class of problems using a similar technique. We also extend our technique to prove similar lower bounds on the time complexity of Random Access Machines; additional arguments are required to show that a RAM cannot use indirect addressing to speed up computations in these cases. In [10], we use some additional tools from approximation theory to prove lower bounds for approximating the square root.

Notice that the gcd problem can be written as an Integer Linear Program. Therefore, one of the consequences of our results is that there is no algorithm for the Integer Linear Programming problem, using operations only from the set $\{+, -, *, /, \text{mod}\}$, whose running time depends only on the number of variables and the number of constraints, and *not* on the absolute value of the coefficients.

Finally, we do not know of any techniques that give nontrivial lower bounds when the set of operations is extended to include all Boolean operations. Finding such a technique would be very interesting.

ACKNOWLEDGMENTS. We are grateful to Michael Sipser for many fruitful discussions. We thank Al Borodin, Shay Kutten, Azaria Paz, and Mike Saks for helpful comments. We would also like to thank Larry Ruzzo for pointing out the work of Stockmeyer [18].

In addition, we are deeply indebted to Éva Tardos for sharing her insights with us, and for drawing our attention to Lemma 2.3.

REFERENCES

1. BABAI, L., JUST, B., AND MEYER AUF DER HEIDE, F. On the limits of computations with the floor function. *Inf. Comput. 78*, 2 (Aug. 1988), 99–107.
2. BEN-OR, M. Lower bounds for algebraic computation trees. In *Proceeding of the 15th ACM Symposium on Theory of Computing*, (Boston, Mass., Apr. 25–27). ACM, New York, 1983, pp. 82–86.
3. BERTONI, A., MAURI, G., AND SABADINI, N. A characterization of the class of functions computable in polynomial time on random access machines. In *Proceedings of the 13th ACM Symposium on Theory of Computing*, (Milwaukee, Wisc., May 11–13). ACM, New York, 1981, pp. 168–176.
4. COLLINS, G. E. The computing time of the Euclidean algorithm. *SIAM J. Comput. 3*, 1 (March 1974), 1–10.
5. DITTERT, E., AND O'DONNELL, M. Lower bounds for sorting with realistic instruction sets. *IEEE Trans. Comput. 34*, 4 (Apr. 1985), 311–317. See also, Correction to: Lower bounds for sorting with realistic instruction sets. *IEEE Trans. Comput. 35*, 10 (Oct. 1986), 932.
6. GRÖTSCHEL, M., LOVÁSZ, L., AND SCHRIJVER, A. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.
7. IBARRA, O. H., MORAN, S., AND ROSIER, L. E. On the control power of integer division. *Theoret. Comput. Sci. 24* (1983), 35–52.
8. JUST, B., MEYER AUF DER HEIDE, F., AND WIGDERSON, A. On computations with integer division. *RAIRO Inform. Theor. Appl. 23*, 1 (1989), 101–111.
9. KNUTH, D. E. *The Art of Computer Programming*, vol. 2. Addison-Wesley, Reading, Mass. 2nd ed., 1981.
10. MANSOUR, Y., SCHIEBER, B., AND TIWARI, P. The complexity of approximating the square root. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science* (Oct.). IEEE, New York, 1989, pp. 325–330.
11. MANSOUR, Y., SCHIEBER, B., AND TIWARI, P. Lower bounds for computations with the floor operation. In *Proceedings of the 16th ICALP*, (Stresa, Italy, July). Lecture Notes in Computer Science, vol. 372. Springer-Verlag, New York, 1989, pp. 559–573. Also *SIAM J. Comput.*, to appear.
12. MORAN, S., SNIR, M., AND MANBER, U. Applications of Ramsey's theorem to decision tree complexity. *J. ACM 32*, 4 (Oct. 1985), 938–949.
13. PAUL, W., AND SIMON, J. Decision trees and random access machines. *Proceedings of the International Symposium held in honor of Ernst Spooker* (Zurich, Switzerland, 1980). In *Monographie de L'Enseigment Mathematique*, No. 30, Université de Geneva, Geneva, Switzerland, 1981, pp. 331–340.
14. PRATT, V. R., AND STOCKMEYER, L. J. A characterization of the power of vector machines. *J. Comput. Syst. Sci. 12* (1976), 198–221.
15. SCHÖNHAGE, A. On the power of random access machines. In *Proceedings of the 6th ICALP* (Graz, Bulgaria, July). Lecture Notes in Computer Science, vol. 71, Springer-Verlag, New York, 1979, pp. 520–529.
16. SHAMIR, A. Factoring numbers in O(log n) arithmetic steps. *Inf. Proc. Lett. 8*, 1 (Jan. 1979), 28–31.
17. SIMON, J. Division in idealized unit cost RAMs. *J. Comput. Syst. Sci. 22* (1981), 421–441.
18. STOCKMEYER, L. Arithmetic versus Boolean operations in idealized register machines. Tech. Rep. RC 5954. IBM T. J. Watson Research Center, Yorktown Heights, N.Y., Apr. 1976.
19. STRASSEN, V. Berechnung und Programm I. *Acta Inf. 1* (1972), 320–335.
20. STRASSEN, V. The computational complexity of continued fractions. *SIAM J. Comput. 12*, 1 (Feb. 1983), 1–27.
21. STEELE, J. M., AND YAO, A. C. Lower bounds for algebraic decision trees. *J. Alg. 3* (1982), 1–8.

RECEIVED NOVEMBER 1988; REVISED DECEMBER 1989; ACCEPTED MARCH 1990