SIGNED b-ADIC PARTITIONS

JAMES M. MANN
Louisiana State University, New Orleans, Louisiana 70122

INTRODUCTION

The common type of partition problem can be stated as follows: let $S \subseteq N$, given $n \in N$, how many ways can we write $n = s_1 + s_2 + \cdots + s_k$, $s_i \in S$? For instance, S might be the squares or the cubes, k might be fixed or not.

This paper considers the question: given b, how many ways can we write $n = a_0 + a_1 b + a_2 b^2 + \cdots + a_m b^m$, $a_i \in \{0, 1, -1, 2, -2, \cdots, b-1, 1-b\}$? An algorithm is derived to answer this question. This algorithm produces for each n a tree, for which questions of height and width are answered.

1. THE DECOMPOSITION ALGORITHM

1.1 Definition. Let $b > 1$ be fixed. A *k-decomposition* of n, $k > 0$, is a partition of n of the form $n = a_0 + a_1 b + a_2 b^2 + \cdots + a_m b^m$, where each $a_i \in \{0, 1, -1, 2, -2, \cdots, b-1, 1-b\}$ and $a_i \neq 0$ for exactly k values of i. A *decomposition* of n is a k-decomposition of n for some (unspecified) k.

The number of k-decompositions of n will be denoted $R_k(n)$. Clearly $R_k(-n) = R_k(n)$, so WLOG we shall assume that $n \geqslant 0$.

1.2 Theorem.

(a) $$R_k(bn) = R_k(n)$$

(b) If $n \equiv a \pmod b$, $a \neq 0$, and if $k > 1$, then

$$R_k(n) = R_{k-1}(n-a) + R_{k-1}(n-a+b)$$

(c) $$R_1(n) = \begin{cases} 1 & \text{if } n = ab^j \text{ for some } j \geqslant 1, \text{ some } 0 < a < b \\ 0 & \text{if } n \neq ab^j \text{ for any } j, \text{ any } a \end{cases}$$

(d) $$R_k(0) = 0 \quad \text{for all } k$$

(e) $$\text{If } 0 < a < b, \text{ then } R_k(a) = 1 \quad \text{for all } k.$$

Proof.

(a) Given any k-decomposition of n, multiplying the expression by b produces a k-decomposition of bn. So $R_k(bn) \geqslant R_k(n)$. Given any k-decomposition of bn, $bn = a_0 + a_1 b + a_2 b^2 + \cdots + a_m b^m$, clearly $b \mid a_0$, so $a_0 = 0$. Dividing the expression by b produces a k-decomposition of n. So $R_k(n) \geqslant R_k(bn)$.

(b) Let $n \equiv a \pmod b$. Consider any k-decomposition of n, $n = a_0 + a_1 b + \cdots + a_m b^m$. $n \equiv a_0 \pmod b$; hence $a \equiv a_0 \pmod b$. Thus either $a = a_0$ or $a = a_0 + b$. That is, the first term of the decomposition is either a or $a - b$. The remaining $k - 1$ terms then are a $(k-1)$-decomposition of $n - a$ or of $n - (a - b)$, respectively.

(c) Immediate from the definition.

(d) Assume false. Then for some k there is at least one k-decomposition of 0, $0 = a_0 + a_1 b + \cdots + a_m b^m$. Place the terms with $a_i < 0$ on the left side of the expression. Then some integer has two distinct representations in base b—contradiction.

(e)
$$\begin{aligned} R_k(a) &= R_{k-1}(a-a) + R_{k-1}(a-a+b) \quad \text{by part (b).} \\ &= 0 + R_{k-1}(1) \quad \text{by parts (d) and (a)} \\ &= R_{k-2}(1-1) + R_{k-2}(1-1+b) = 0 + R_{k-2}(1) \\ &= \cdots = R_1(1) \\ &= 1 \quad \text{by part (c).} \end{aligned}$$

This theorem enables us quickly to find $R_k(n)$. Moreover, unwinding the algorithm, we can find the k-decompositions.

Example 1. Let $b = 4$.

$$R_5(3) = R_4(0)+R_4(4) = 0+R_4(1) = R_3(0)+R_3(4) = R_3(1) = R_2(0)+R_2(4) = R_2(1) = R_1(0)+R_1(4) = 1,$$

a result we know already. Unwinding the algorithm,

$$4 = 4,\quad 1 = -3+4,\quad 4 = -12+16,\quad 1 = -3-12+16,\quad 4 = -12-48+64,$$
$$1 = -3-12-48+64,\quad 4 = -12-48-192+256,$$
$$3 = -1-12-48-192+256 = -1-3\cdot 4-3\cdot 4^2-3\cdot 4^3+1\cdot 4^4 .$$

The pattern is clear, so from now on we shall use part (e) of the theorem and stop the algorithm whenever the argument n is less than b. Moreover, because of part (a), we shall consider only n such that b does not divide n.

Example 2. Let $b = 3$.

$$R_4(17) = R_3(15)+R_3(18) = R_3(5)+R_3(2) = R_2(3)+R_2(6)+R_3(2) = R_2(1)+R_2(2)+R_3(2) = 1+1+1 = 3.$$

Unwinding,

$1 = -2+3$	$2 = -1+3$	$2 = -1-6+9$
$3 = -6+9$	$6 = -3+9$	$18 = -9-54+81$
$5 = 2-6+9$	$5 = -1-3+9$	$17 = -1-9-54+81$
$15 = 6-18+27$	$15 = -3-9+27$	
$17 = 2+6-18+27$	$17 = 2-3-9+27$	
$= 2+2\cdot 3-2\cdot 3^2+1\cdot 3^3$		

Example 3. Let $b = 2$.

$$R_3(11) = R_2(10)+R_2(12) = R_2(5)+R_2(3) = R_1(4)+R_1(6)+R_1(2)+R_1(4) = 1+0+1+1 = 3.$$

Unwinding,

$4 = 4$	$2 = 2$	$4 = 4$
$5 = 1+4$	$3 = 1+2$	$3 = -1+4$
$10 = 2+8$	$12 = 4+8$	$12 = -4+16$
$11 = 1+2+8$	$11 = -1+4+8$	$11 = -1-4+16$

1.3. Each time k decreases by one, each term $R_k(\cdot)$ splits into at most two terms $R_{k-1}(\cdot)$. In completing the algorithm, there are $k-1$ such steps. Hence $R_k(n) \leqslant 2^{k-1} < 2^k$ for all n. We have the well known result

Theorem. $\{b^i : i = 0, 1, 2, \cdots\}$ is a Sidon set. (See [2], pp. 124, 127.)

1.4 Lemma. If $n = a_0 + a_1 b + a_2 b^2 + \cdots + a_m b^m$ is any decomposition of n, $a_m \neq 0$, then $a_m > 0$.

Proof. If $a_m < 0$, then

$$n = \sum_{i=0}^{m-1} a_i b^i + a_m b^m \leqslant \sum_{i=0}^{m-1} (b-1)b^i - b^m = b^m - 1 - b^m = -1$$

—a contradiction.

1.5 Definition. A k-decomposition of n is *basic* if (a) $a_m > 1$, or if (b) $a_{m-1} \geqslant 0$ (or both).

Theorem. Let $b^{h-1} < n < b^h$. Then for any basic decomposition of n,

(a) $i > h \Rightarrow a_i = 0$

(b) $0 \leqslant a_h \leqslant 1$

(c) If $a_h = 0$, then $a_{h-1} > 0$

(d) If $a_h = 1$, then $a_{h-1} = 0$; and if $a_j b^j$ is the last non-zero term before $a_h b^h$, then $a_j < 0$.

Proof. (a) By the lemma above, if $a_m b^m$ is the last non-zero term, $a_m > 0$. Assume $m > h$.

Case 1. $a_m > 1$. Then

$$n = \sum_{i=0}^{m} a_i b^i \geqslant \sum_{i=0}^{m-1} (1-b)b^i + 2b^m = b^m + 1 > b^h$$

—a contradiction.

Case 2. $a_m = 1$ and $a_{m-1} \geqslant 0$. Then

$$n \geqslant \sum_{i=0}^{m-2} (1-b)b^i + 0b^{m-1} + b^m = 1 + b^{m-1}(b-1) \geqslant 1 + b^{m-1} \geqslant 1 + b^h$$

—a contradiction.

(b) By part (a), there are no terms in the decomposition after $a_h b^h$, so $a_h \geqslant 0$. Assume $a_h > 1$. Then

$$n \geqslant \sum_{i=0}^{h-1} (1-b)b^i + 2b^h = 1 + b^h$$

—a contradiction.

(c) If $a_h = 0$, then there are no terms after $a_{h-1}b^{h-1}$, so $a_{h-1} \geqslant 0$. Assume $a_{h-1} = 0$. Then

$$n \leqslant \sum_{i=0}^{h-2} (b-1)b^i = b^{h-1} - 1$$

—a contradiction.

(d) If $a_h = 1$, then by the definition of a basic decomposition $a_{h-1} \geqslant 0$. Assume $a_{h-1} > 0$. Then

$$n \geqslant \sum_{i=0}^{h-2} (1-b)b^i + 1b^{h-1} + 1b^h = 1 + b^h$$

—a contradiction. The same reasoning shows that if the next to last non-zero coefficient is a_j, $j < h$, then $a_j < 0$.

Corollary. Let $b^{h-1} < n < b^h$, and let $k > h$. Then no k-decomposition of n is basic.

Proof. Every basic decomposition of n ends with $a_{h-1}b^{h-1}$ or with $a_{h-2}b^{h-2} + 0 \cdot b^{h-1} + 1 \cdot b^h$. In either case there are at most h non-zero terms in the sum.

1.6 Theorem. Starting with $R_k(a)$, $0 < a < b$, the unwinding of the algorithm produces a basic decomposition of n iff $k = 1$.

Proof. Start with a k-decomposition of a.

Case 1. $k = 1$. The reverse algorithm starts: $x_1 = a$; then $x_2 = ab^p$, $p \geqslant 1$; then $x_3 = ab^p + a'$.

Case 1a. $a > 1$ or $p > 1$. Then a' can be any integer such that $0 < |a'| < b$.

Case 1b. $a = p = 1$. Then $x_3 = b + a'$. If $a' < 0$, $x_3 < b$. But the forward algorithm stops as soon as the argument is less than b. So $a' > 0$. In either case there is a basic 2-decomposition of x_3. The next step is to multiply by b^q for some $q \geqslant 1$. Clearly the resulting 2-decomposition is basic. Then add a''; the new 3-decomposition is still basic. Continue until a basic decomposition of n is reached.

Case 2. $k > 1$. By the corollary above, since $a < b^1$, no k-decomposition of a is basic. That is, the reverse algorithm starts

$$a = a_0 + a_1 b + \cdots + a_{m-1}b^{m-1} + b^m,$$

with $a_{m-1} < 0$. Multiplying by b^p produces a non-basic k-decomposition. Then adding a' gives a non-basic $(k+1)$-decomposition. Continue, ending with a non-basic decomposition of n.

1.7 Definition. Let $B_k(n)$ be the number of basic k-decompositions of n. Let

$$B(n) = \sum_{k=1}^{\infty} B_k(n).$$

Remark. Since $n < b^h$, $k > h \Rightarrow B_k(n) = 0$ (corollary above), the sum is only finite.

Theorem. If $b^{h-1} < n < b^h$, $k > h$, then $R_k(n) = R_h(n) = B(n)$; and $B(n) \leqslant 2^{h-1}$.

Proof. If $k > h$, no k-decomposition of n is basic. Thus the algorithm goes all the way: every end term is of the form $R_s(a)$, $0 < a < b$, $s > 1$. Once all the $a < b$ appear, no more decompositions can appear. Each basic decomposition occurs from unwinding each $R_1(a)$, choosing $k \leqslant h$ so that $s = 1$ when the a first appears. The inequality is from 1.3.

2. THE CASE $b = 2$

From the algorithm, we see that if neither n nor $n+1$ is divisible by b, then their k-decompositions differ only in the first term. Therefore, for simplification we shall assume that $b = 2$, unless specifically stated otherwise. Of course, we restrict n to be odd.

2.1 By the algorithm, $R_k(n) = R_{k-1}(n-1) + R_{k-1}(n+1)$. Let $n - 1 = 2^p x$ and $n + 1 = 2^q y$, x and y odd. Note that $min\ (p,q) = 1$ and that $max\ (p,q) \geq 2$, as $n-1$ and $n+1$ are consecutive even integers.

Definition. Given x, y odd, if there exists an (odd) n such that $R_k(n) = R_{k-1}(x) + R_{k-1}(y)$, write $x * y = n$. If no such n exists, then $x * y$ is undefined.

Remark. By the uniqueness of the algorithm,

(a) $x * y = y * x$ if either exists, and

(b) $x * y = u * v \Rightarrow \{x,y\} = \{u,v\}$.

2.2 Theorem. Let y be given. If $x \geq y$, then $x * y$ exists iff $x = 2^i y + 1$ or $x = 2^i y - 1$ for some $i \geq 1$. If so, then $x * y = 2^{i+1}y + 1$ or $x * y = 2^{i+1}y - 1$, respectively.

Proof. By the algorithm, if $x * y$ is to exist, there must exist p, $q \geq 1$ such that $2^p x - 2^q y = \pm 2$. By the note above, $p = 1$ and $q \geq 2$. So $x = 2^{q-1}y \pm 1$. Let $i = q - 1 \geq 1$. $x * y$ is the odd integer between $2x$ and $2^q y$. So

$$x * y = (\tfrac{1}{2})[2x + 2^q y] = (\tfrac{1}{2})[2(2^i y \pm 1) + 2^{i+1}y] = 2^{i+1} \pm 1.$$

Corollary. If GCD $(x,y) > 1$, then $x * y$ does not exist. In particular, if $y > 1$, then $y * y$ does not exist.

2.3 Theorem. $3 * 1 = \{5,7\}$. In all other cases, $x * y$ is unique.

Proof. WLOG $x \geq y$. If $x * y$ exists, $x = 2^i y \pm 1$. If $x * y$ is not unique, then x must be expressible in two ways, i.e.,

$$x = 2^p y + 1 = 2^q y - 1$$

for some p, $q \geq 1$. Then

$$2^q y - 2^p y = 2, \qquad 2^{q-1}y - 2^{p-1}y = 1.$$

Since y divides the left side, $y = 1$. Then $p = 1$ and $q = 2$. So

$$x = 2^1 \cdot 1 + 1 = 3 = 2^2 \cdot 1 - 1, \quad \text{and} \quad x * y = 2^2 \cdot 1 + 1 = 5, \qquad x * y = 2^3 \cdot 1 - 1 = 7.$$

2.4 Theorem. Given $x > 3$, there exist two y, $y < x$, such that $x * y$ exists.

Proof. $x = 2^i y \pm 1$, so $y = (x-1)/2^p$ and $y = (x+1)/2^q$, y odd. These numbers are distinct unless $(x-1)/2^p = (x+1)/2^q$. If so, then since $x - 1$, $x + 1$ are consecutive even numbers, both divisible by some power of 2, $x = 3$.

Corollary. If $a * b$ exists, then the integers y, $y < a * b$, such that $(a * b) * y$ exists are $y = a$ and $y = b$.

Proof. If $a * b$ exists, WLOG $a \geq b$. Then $a = 2^i b \pm 1$. By the theorem, if $(a * b) * y$ exists, then

$$y = \frac{(a*b) \mp 1}{2^p} = \frac{(2^{i+1}b \pm 1) \mp 1}{2^p} = \left\{ \frac{2^{i+1}b}{2^p}, \frac{2^{i+1}b \pm 2}{2^q} \right\}$$

$$= \{b, 2^i b \pm 1\} = \{b, a\}.$$

Remark. If $a = b$, by the Corollary of 2.2, $a = b = 1$, and so $y = 1$.

2.5 Theorem. If $x * y$ exists, then exactly one of $\{x, y, x * y\}$ is divisible by 3.

Proof. $1 * 1 = 3$. Assume now WLOG that $x > y$. So $x = 2^i y \pm 1$, and $x * y = 2^{i+1}y \pm 1$.

Case 1. Clearly if $3|y$, 3 divides neither x nor $x * y$.

Case 2. If $3|x$, 3 cannot divide y. Assume $3|x * y$. Then $3|(x * y - x)$, so

$$3|(2^{i+1}y - 2^i y),\ 3|2^i y \text{ — a contradiction.}$$

Case 3. Assume that 3 divides neither x nor y. To show $3|x * y$.

Case 3a. $y \equiv 1 \pmod 3$. Since $2^i \equiv (-1)^i \pmod 3$,

$$x = 2^i y \pm 1 \equiv (-1)^i \pm 1 \pmod 3.$$

Since $x \not\equiv 0 \pmod 3$, if i is even, we must use the $+1$, and if i is odd, we must use the -1. Then

$$x * y = 2^{i+1}y \pm 1 \equiv (-1)^{i+1} \pm 1 \pmod 3, \qquad x * y \equiv 0 \pmod 3$$

whether i is even or odd.

Case 3b. $y \equiv -1 \pmod 3$. Then

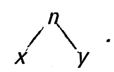
$$x \equiv (-1)^{i+1} \pm 1 \pmod 3.$$

If i is even, we use the -1; if i is odd, the $+1$. So

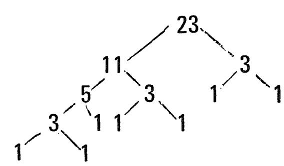
$$x * y \equiv (-1)^{i+2} \pm 1 \pmod 3 \equiv 0 \pmod 3$$

in both cases.

2.6. The expression $n = x * y$ can conveniently be expressed visually as



If x or $y > 1$, it in turn can be written as a $*$-product. Each n has in this manner associated with it a tree. For example, for $n = 23$, the tree is as in Fig. 1.



$$H(23) = 5, \quad W(23) = 7$$

Figure 1

Remark. Since $x, y < x * y$, the numbers decrease down the tree, and every chain ends with 1. The tree associated with n, without integers at the nodes, with the longer chain always to the left at every node, will be denoted $T(n)$.

2.7 Definition. If the length of the longest chain in the tree is ℓ, then the *height* of the tree, denoted $H(n)$, is defined by $H(n) = \ell + 1$. The number of branches of the tree (= number of times 1 appears) is the *width* of the tree, denoted by $W(n)$.

Lemma. Let $n = x * y$, $x \geqslant y$. Then

(a) $$H(n) = 1 + H(x)$$

(b) $$W(n) = W(x) + W(y).$$

Proof. Obvious from the definition of $T(n)$.

Theorem. Let $2^{h-1} < n < 2^h$. Then

(a) $$H(n) = h$$

(b) $$W(n) = B(n), \text{ the number of basic decompositions}$$

(c) $$h \leqslant W(n) \leqslant 2^{h-1}$$

Proof. (a) If $h = 1$, $H(1) = 1$; if $h = 2$, $H(3) = 2$. Assume that for all $n < 2^k$, the statement is true. Let $2^k < n < 2^{k+1}$. The algorithm starts: $R_s(n) = R_{s-1}(n-1) + R_{s-1}(n+1)$.

Case 1. $n - 1$ is divisible by 4. Then $n + 1$ is not divisible by 4, so $2^k < n + 1 < 2^{k+1}$. $2^{k-1} < (n+1)/2 < 2^k$. By the inductive hypothesis, $H((n+1)/2) = k$. By the lemma, $H(n) = k + 1$.

Case 2. $n + 1$ is divisible by 4. Then $2^k < n - 1 < 2^{k+1}$; $2^{k-1} < (n-1)/2 < 2^k$. So $H((n-1)/2) = k$; $H(n) = k + 1$.

(b) The algorithm produces the numbers at the nodes of the tree. As soon as a 1 appears, the branch stops. Starting with $R_1(1)$, following each chain upwards produces each of the basic decompositions.

(c) The second inequality is the Theorem of 1.7. The first is obvious for $n = 1, 3$. Assume the first inequality is true for all $n < 2^k$. Let $2^k < n < 2^{k+1}$. $n = x * y$ for some $x > y$, $2^{k-1} < x < 2^k$. By the inductive hypothesis, $W(x) \geqslant k$. So $W(n) = W(x) + W(y) \geqslant k + 1$.

2.8 Lemma. Let $0 < t < 2^{h-1}$, t odd. Then $T(2^{h-1} + t) = T(2^h - t)$.

Proof. If $h = 2$, then $t = 1$. $2^{2-1} + 1 = 3 = 2^2 - 1$; the result is automatically true. If $h = 3$, then $t = 1$ or 3. $2^{3-1} + 1 = 5$ and $2^3 - 1 = 7$; while $2^{3-1} + 3 = 7$ and $2^3 - 3 = 5$. We know $T(5) = T(7)$.

Assume that the statement is true for all $k \leqslant h$. Let t be any odd number such that $0 < t < 2^k$. If $2^k + t = 2^{k+1} - t$, then $t = 2^{k-1}$; since t is odd, $t = k = 1$.

Case 1. $t+1$ is divisible by 4. Then

$$2^k+t = \frac{2^k+t+1}{2^p} * \frac{2^k+t-1}{2},$$

where 2^p is the highest power of 2 that divides $t+1$, $2 \leqslant p \leqslant k$.

$$= \left(2^{k-p}+\frac{t+1}{2^p}\right) * \left(2^{k-1}+\frac{t-1}{2}\right);$$

and

$$2^{k+1}-t = \frac{2^{k+1}-(t+1)}{2^p} * \frac{2^{k+1}-(t-1)}{2} = \left(2^{k-p+1}-\frac{t+1}{2^p}\right) * \left(2^k-\frac{t-1}{2}\right).$$

By the inductive hypothesis,

$$T\left(2^{k-p}+\frac{t+1}{2^p}\right) = T\left(2^{k-p+1}-\frac{t+1}{2^p}\right)$$

and

$$T\left(2^{k-1}+\frac{t-1}{2}\right) = T\left(2^k-\frac{t-1}{2}\right).$$

Thus $T(2^k+t)$ and $T(2^{k+1}-t)$ have the same right branch, the same left branch, and therefore are equal.

Case 2. $t-1$ is divisible by 4. Interchange $t-1$, $t+1$ in the above proof.

Theorem. If $h \geqslant 3$, there are 2^{h-3} different trees of height h associated with the odd integers.

Proof. For $h=3$, $T(5)=T(7)$, so there is one tree of height 3. Let $k \geqslant 3$. To each x, $2^{k-1} < x < 2^k$ there exist $y_1 \neq y_2$, $y_i < x$, such that $x * y_i$ exists. Since $H(y_1) \neq H(y_2)$, $T(x*y_1) \neq T(x*y_2)$. Therefore the number of trees of height $k+1$ is at least twice the number of trees of height k. Hence the number of trees of height h is at least 2^{h-3}.

Between 2^{h-1} and 2^h there are 2^{h-2} odd integers. By the lemma, each tree of height h is associated with at least two integers. Hence the number of trees of height h is at most 2^{h-3}.

2.9 Theorem. $W(2^{h-1}+1) = W(2^h-1) = h$; the minimum possible width of a tree of height h is attained.

Proof. If $h=3$, $W(2^{3-1}+1) = W(5) = 3$. Assume that $W(2^{k-1}+1) = k$.

$$2^k+1 = (2^{k-1}+1)*1.$$

It follows that

$$W(2^k+1) = W(2^{k-1}+1)+W(1) = k+1.$$

Since $W(n) \geqslant h$ if $2^{h-1} < n < 2^h$, the minimum width is attained. Lastly, by the lemma above, $W(2^h-1) = h$.

Theorem. (a) The maximum width of any tree of height h is F_{h+1}, where F_i is the i^{th} Fibonacci number.

(b) This width is attained for

$$n = (2^{h+1}+(-1)^h)/3, \quad h \geqslant 1,$$

and for

$$n = (5\cdot 2^{h-1}+(-1)^{h-1})/3, \quad h \geqslant 2.$$

Proof. For $h=1$, $W(1)=1$. For $h=2$, $W(3)=2$. For $h=3$, $W(5)=W(7)=3$.

(a) For each k, the maximum width is attained by at least two values of n. Call the smallest of these values n_k, i.e., $\{n_k\} = \{1, 3, 5, 11, \cdots\}$. Assume:

(1) $$W(n_i) = F_{i+1}, \qquad i = 1, 2, \cdots, k$$

(2) $n_k = n_{k-1} * n_{k-2}$. The two inductive hypotheses are true for $k=3$. By the Corollary of 2.4, $n_k * n_{k-1} = n$ exists; so

$$W(n) = W(n_k)+W(n_{k-1}) = F_{k+1}+F_k = F_{k+2}.$$

$T(n)$ has as its left branch the widest tree of height k, as its right branch the widest tree of height $k-1$. Hence $T(n)$ is the widest tree of height $k+1$, and there is only one such tree. Since n is the smaller integer whose tree has this width, $n = n_{k+1}$.

(b) Claim: $n_h = 2n_{h-1}+(-1)^h$. Statement is true for $h=2$. Assume it is true for $h=k$. Then $2n_k = 4n_{k-1}+2(-1)^k$. Using the algorithm, we can calculate $n_{k+1} = n_k * n_{k-1}$. Since $2n_k$ and $4n_{k-1}$ differ by 2,

$$n_{k+1} = (\tfrac{1}{2})[2n_k + 4n_{k-1}] = (\tfrac{1}{2})[2n_k + 2n_k - 2(-1)^k] = 2n_k + (-1)^{k+1} .$$

Claim proved. Assume

$$n_k = \frac{2^{k+1} + (-1)^k}{3} .$$

By the claim,

$$n_{k+1} = 2\left(\frac{2^{k+1} + (-1)^k}{3}\right) + (-1)^{k+1} = \frac{2^{k+2} + (-1)^{k+1}}{3} .$$

Lastly, if m_h is the larger number such that $W(m_h) = F_{h+1}$, by the Lemma of 2.8, $m_h + n_h = 2^{h-1} + 2^h$. So

$$m_h = 3 \cdot 2^{h-1} - n_h = \frac{5 \cdot 2^{h-1} + (-1)^{h-1}}{3} .$$

Theorem. If the base is $b > 2$, then $W((b^h - 1)/(b - 1)) = 2^{h-1}$; that is, the maximum width attained is the maximum possible.

Proof. It is clear that $W(b + 1) = W(b + 2) = 2$. Assume that $W(m) = W(m + 1) = 2^{k-1}$ where $m = (b^k - 1)/(b - 1)$.

$$m * (m + 1) = \left\{ bm + 1, bm + 2, \cdots, bm + b - 1 \right\}$$

(from the obvious definition of $x * y$, $\left\{ x * y \right\}$ has at least $b - 1$ elements.) So

$$W(bm + 1) = W(bm + 2) = W(m) + W(m + 1) = 2^k \quad \text{and} \quad bm + 1 = b\left(\frac{b^k - 1}{b - 1}\right) + 1 = \frac{b^{k+1} - 1}{b - 1} .$$

Remark. Comparison of the preceding two theorems shows why the special case $b = 2$ is more interesting than the general case. The trees for $b = 2$ are of special type: at any node the two sub-trees are always of unequal heights.

3. THE PROBLEM OF WIDTHS

3.1 Theorem. $2 | W(n)$ iff $3 | n$.

Proof. $W(1) = 1$ and $W(3) = 2$. Assume the statement is true for all $n \leq k$. Consider $W(k + 1)$. Let $k + 1 = x * y$.

Case 1. $k + 1$ is divisible by 3. By the Theorem of 2.5, neither x nor y is divisible by 3. By the inductive hypothesis $W(x)$ and $W(y)$ are odd. Hence $W(k + 1) = W(x) + W(y)$ is even.

Case 2. $k + 1$ is not divisible by 3. Then one of x,y is. So $W(k + 1) =$ even + odd = odd.

3.2. An interesting but unsolved question is the following: given w, find all (odd) n such that $W(n) = w$.

If $n > 2^w$, then $H(n) > w$, so $W(n) > w$ (Theorem of 2.7). Thus all solutions n satisfy $n < 2^w$. At least one pair of solutions always exists, because

$$W(2^{w-1} + 1) = W(2^w - 1) = w$$

(first Theorem of 2.9). From the theorem above it appears that there should be fewer solutions for w even than for w odd. An examination of a short table of solutions, found by the algorithm, shows little regularity.

REFERENCES

1. G.H. Hardy and E.M. Wright, *An Introduction to the Theory of Numbers,* 4th ed., The Clarendon Press, Oxford, 1960.
2. Walter Rudin, *Fourier Analysis on Groups,* Interscience, New York, 1962.

****