One Pile Nim with Arbitrary Move Function

by Arthur Holshouser and Harold Reiter

Arthur Holshouser
3600 Bullard St.
Charlotte, NC,
USA, 28208

Harold Reiter
Department of Mathematics
UNC Charlotte
Charlotte, NC 28223

Introduction: The purpose of this paper is to solve a class of combinatorial games consisting of one-pile counter pickup games for which the maximum number of counters that can be removed on each successive move changes during the play of the game.

Two players alternate removing a positive number of counters from the pile. An ordered pair (N, x) of positive integers is called a position. The number N represents the size of the pile of counters, and x represents the greatest number of counters that can be removed on the next move. A function $f : Z^+ \to Z^+$ is given which determines the maximum size of the next move in terms of the current move size. Thus a move in a game is an ordered pair of positions $(N, x) \mapsto (N-k, f(k))$, where $1 \leq k \leq \min(N, x)$.

The game ends when there are no counters left, and the winner is the last player to move in a game. In this paper we will consider $f : Z^+ \to Z^+$ to be completely arbitrary. This paper extends a previous paper by the authors [5], which in turn extended two other papers, [3] and [6]. The paper by Epp, Ferguson [3] assumed f is non-decreasing, and the paper by Schwenk [6] assumed f is non-decreasing and $f(n) \geq n$. Our previous paper [5] assumed more general conditions of f including as a special case all $f : Z^+ \to Z^+$ that satisfy $f(n+1) - f(n) \geq -1$.

The main theorem of this paper will also allow the information concerning the strategy of a game to be stored very efficiently, and our paper [5] is a subcase of this paper. We now proceed to develop the theory.

Generalized Bases: An infinite strictly increasing sequence $B = (b_0 = 1, b_1, b_{2,} \cdots)$ of positive integers is called an infinite g-base if for each $k \geq 0, b_{k+1} \leq 2b_k$. This 'slow growth' of B's members guarantees lemma 1.

Finite g-bases. A finite strictly increasing sequence $B = (b_0 = 1, b_1, b_2, \cdots, b_t)$ of positive integers is called a *finite g-base* if for each $0 \leq k < t, b_{k+1} \leq 2b_k$.

Lemma 1 *Let B be an infinite g-base. Then each positive integer N can be represented as $N = b_{i_1} + b_{i_2} + \cdots + b_{i_t}$ where $b_{i_1} < b_{i_2} < \cdots < b_{i_t}$ and each b_{i_j} belongs to B.*

Proof. The proof is given by the recursive algorithm. Note that $b_0 = 1 \in B$. Suppose all integers $1, 2, 3, \cdots, m-1$ have been represented as a sum of distinct members of B by the algorithm. Suppose $b_k \leq m < b_{k+1}$. Then $m = (m - b_k) + b_k$. Now $m - b_k < b_k$, for otherwise, $2b_k \leq m$. Since $m < b_{k+1}$, we have $2b_k < b_{k+1}$, which contradicts the definition of a g-base. Since $m - b_k$ is less than m, it follows that $m - b_k$ has been represented by the algorithm as a sum of distinct members of B that are less than b_k. Thus we may assume that $m - b_k = b_{i_1} + b_{i_2} + \cdots + b_{i_{t-1}}$ where $b_{i_1} < b_{i_2} < \cdots < b_{i_{t-1}}$ and each b_{i_j} belongs to B. Then $m = b_{i_1} + b_{i_2} + \cdots + b_{i_t}$ where $b_{i_t} = b_k, b_{i_1} < b_{i_2} < \cdots < b_{i_t}$ and each b_{i_j} belongs to B. ■

Lemma 2 *Let $B = (b_0 = 1, b_1, b_2 \cdots b_t)$ be a finite g-base. For any positive integer N, let $\theta \geq 0$ be the unique integer such that $0 \leq N - \theta b_t < b_t$. Then the same* *algorithm used in the proof of lemma 1 can be used to uniquely represent* $N = b_{i_1} + b_{i_2} + \cdots + b_{i_k} + \theta b_t$ *where* $b_{i_1} < b_{i_2} < \cdots < b_{i_k} < b_t$ *and each* b_{i_j} *belongs to* B.

In this paper we always use the algorithms used in the proofs of lemmas 1 and 2 to uniquely represent any positive integer N as the sum of distinct members of the g-base that we are dealing, whether this g-base is finite or infinite.

Definition 3 *Suppose* $B = (b_0 = 1, b_1, b_2, \cdots)$, *where* $b_0 < b_1 < b_2 < \cdots$, *is an infinite* g*-base. Let* $N = b_{i_1} + b_{i_2} + \cdots + b_{i_k}$, *where* $b_{i_1} < b_{i_2} < \cdots < b_{i_k}$, *be the representation of positive integer* N *that is specified by the algorithm used in the proof of lemma 1. Then we define* $\overline{g}(N) = b_{i_1}$.

Suppose $B = (b_0 = 1, b_1, b_2, \cdots b_t)$, where $b_0 < b_1 < \cdots < b_t$, is a finite g-base. Let $N = b_{i_1} + b_{i_2} + \cdots + b_{i_k} + \theta b_t$, where $b_{i_1} < b_{i_2} < \cdots < b_{i_k} < b_t$ be the representation of N in B that is specified by the algorithm used in lemma 2. Then $\overline{g}(N) = b_{i_1}$ unless $N = \theta b_t$ in which case $\overline{g}(N) = b_t$.

Generating g-bases: For every function $f : Z^+ \to Z^+$, we generate a g-base B_f and a function $g' : B_f \to Z^+$ as follows.

Let $b_0 = 1$, $g'(b_0) = 1, b_1 = 2$, $g'(b_1) = 2$.

Suppose $b_0, b_1, b_2, \cdots, b_k$ and $g'(b_0)$, $g'(b_1)$, $g'(b_2), \cdots, g'(b_k)$, where $k \geq 1$, have been generated. Then $b_{k+1} = b_k + b_i$ where b_i is the smallest member of $\{b_0, b_1, \cdots, b_k\}$ such that $g'(b_i) = b_i$ and $f(b_i) \geq g'(b_k)$ if such a b_i exists. If no such b_i exists for some k, the g-base B_f is finite. Also, $g'(b_{k+1}) =$

$\min[\{b_{k+1}\} \cup \{b_{k+1} - b_k + \overline{x} : 1 \leq \overline{x} < b_k \text{ and } f(b_{k+1} - b_k + \overline{x}) < g'(\overline{g}(b_k - \overline{x}))\}]$. Of course, $\overline{g}(b_k - \overline{x})$ is computed using definition 1 with $(b_0 = 1, b_1, b_2, \cdots, b_k)$, and $\min S$ means the smallest member of S. We will explain later why B_f and g' are defined this way.

Definition 4 *Suppose $f : Z^+ \to Z^+$ generates the g-base B_f and the function $g' : B_f \to Z^+$. Then for every $N \in Z^+$, we define $g'(N) = g'(\overline{g}(N))$, where $\overline{g}(N)$ is computed using B_f . Thus in the definition of $g'(b_{k+1})$, we could substitute $g'(b_k - \overline{x})$ for $g'(\overline{g}(b_k - \overline{x}))$.*

Before we state the main theorem, we need a few more definitions. We recall that for our game, we are given some arbitrary function $f : Z^+ \to Z^+$. Also, a position in the game is an ordered pair of positive integers (N, x), and a move is $(N, x) \mapsto (N - k, f(k)), 1 \leq k \leq \min(N, x)$. For each ordered pair (N, x), define $F(N, x) = 0$ if (N, x) is a safe position, and $F(N, x) = 1$ if (N, x) is an unsafe position. Note that $F(N, x) = 1$ when the list $F(N - 1, f(1)), F(N - 2, f(2)), \cdots, F(N - x, f(x))$ contains at least one 0. Note that $F(0, x) = 0$ for all $x \in Z^+$. $F(N, x) = 0$ when this list contains no 0's. We imagine that $F(1, x), x = 1, 2, \cdots$, is computed first. Then $F(2, x), x = 1, 2, 3, \cdots$, is computed. Then $F(3, x), x = 1, 2, 3, \cdots$, is computed, etc. Note that $F(N, x) = 1$ when $N \leq x$. Note also that for a fixed $N \in Z^+$ and a variable $x \in Z^+$, the infinite sequence $F(N, x), x = 1, 2, 3, \cdots$, always consists of a finite string (possibly empty) of consecutive 0's followed by an infinite string of consecutive 1's. This is because $F(N, x) = 1$ when $x \geq N$ and also once the sequence first switches from 0 to 1 it must retain the value of 1 thereafter. For each $N \in Z^+$, define $g(N)$ to be the smallest $x \in Z^+$ such that $F(N, x) = 1$. Of course $1 \leq g(N) \leq N$. For every $N \in Z^+, g(N)$ is the position of the first 0 in the sequence $F(N-1, f(1)), F(N-2, f(2)), \cdots, F(N-g(N), f(g(N))$. This means that $F(N-g(N), f(g(N))) = 0$, but all preceding members of this sequence have a value of 1. It is obvious that $F(N, x) = 0$ when $x \leq g(N) - 1$ and $f(N, x) = 1$ when $x \geq g(N)$. We can now state the main theorem.

Main theorem: Suppose we play our game with an arbitrary but fixed move function $f : Z^+ \to Z^+$. Suppose f generates the g-base B_f and the function $g' : B_f \to Z^+$. Then for every $N \in Z^+, g(N) = g'(N)$ where $g'(N)$ is defined in definition 4 using the g-base B_f and the function $g' : B_f \to Z^+$.

The main theorem implies that a position (N, x) is unsafe if $x \geq g'(N)$ and safe if $x < g'(N)$. This means a winning move must be $(N, x) \to (N - g'(N), f(g'(N)))$. The reader will note that the theorem is true whether B_f is finite or infinite. In a moment we will write a detailed proof for the case where B_f is infinite. The proof of the finite case, which involves a slight modification of the infinite case, is left to the reader.

Before we begin the proof, we would like to point out that for an enormous number of functions f it is very easy to compute B_f and g'. For example, if f is non-decreasing or if f satisfies $f(n+1) - f(n) \geq -1$, then $g' : B_f \to Z^+$ is just the identity function on B_f, and B_f is generated by the following very simple algorithm. $b_0 = 1, b_1 = 2$ and if $b_0, b_1, b_2, \cdots, b_k$ have been generated, then $b_{k+1} = b_k + b_i$ where b_i is the smallest member of $\{b_0, b_i, \cdots b_k\}$ such that $f(b_i) \geq b_k$. There are also many other functions f for which B_f and g' are very easy to compute. However, for many functions f, the best way to compute B_f and g' is to go ahead and compute $g(N)$ first. Note that $g(N), \quad N = 1, 2, 3, \cdots$, can be computed directly by the following algorithm: $g(1) = 1, g(2) = 2$. If $g(1), g(2), \cdots, g(k-1), k \geq 3$, have been computed, then $g(k)$ is the smallest $x \in \{1, 2, 3, \cdots, k\}$ such that $f(x) < g(k - x)$, where we agree that $g(0) = \infty$.

For those functions where $g(N)$ is computed first, it may appear that this paper has no advantages whatsoever since we already know $g(N)$. However, once $g(N)$ is computed, we know that $g' = g$, and we can then easily compute B_f.

Once we know B_f and $g' : B_f \to Z^+$, we know that B_f and g' by themselves store the complete strategy of the game. Quite often the members of B_f grow exponentially. In these cases we have an efficient way of storing the strategy of the game. We conjecture that if g is unbounded, then the members of B_t always grow exponentially.

At the end of this paper, we give two examples in which B_f, g' provide efficient storage. We will also give an example when B_f, g' provide inefficient storage. In the first two examples, g is unbounded and in the third, g is bounded.

Proof. The main theorem follows easily if we can prove the following statements.

We assume B_f is infinite.

1. $\forall b_i \in B_f, g(b_i) = g'(b_i)$,

2. $\forall N \in Z^+ \backslash B_f, g(N) < N$,

3. $\forall N \in Z^+ \backslash B_f$, if $b_i < N < b_{i+1}$, then $N = b_i + (N - b_i)$, where $1 \le N - bi < b_i$, and $g(N) = g(N - b_i)$. Of course, $1 \le N - b_i < b_i$ is obvious since B_f is a g-base.

Note: At the end of the proof, we will use property 3 to explain why we defined B_f the way that we did.

The main theorem follows because if $N = b_{i_1} + b_{i_2} + \cdots + b_{i_k}$, where $b_{i_1} < b_{i_2} < \cdots < b_{i_k}$, is the representation of N in the infinite g-base B_f that is computed by the algorithm used in the proof of lemma 1, we have the following.

$$\begin{aligned} g(N) &= g((b_{i_1} + \cdots + b_{i_{k-1}}) + b_{i_k}) = g((b_{i_1} + \cdots + b_{i_{k-2}}) + b_{i_{k-1}}) \\ &= g((b_{i_1} + \cdots + b_{i_{k-3}}) + b_{i_{k-2}}) = \cdots = \\ &= g(b_{i_1}) = g'(b_{i_1}) = g'(N), \end{aligned}$$

by the definition of $g'(N)$ since $g(b_{i_1}) = g'(\overline{g}(N)$.

Note that once we have proved statements $1, 2, 3$ for all members of $\{1, 2, 3, \cdots, \overline{k}\}$, then $\forall N \in \{1, 2, 3, \cdots \overline{k}\}$, $g(N) = g'(N)$ will also be true.

We prove statements $1, 2, 3$ by mathematical induction. First, note that no matter what f is $g(1) = g'(1) = 1$ and $g(2) = g'(2) = 2$. Conditions $2, 3$ do not apply for integers $1, 2$ since $\{1, 2\} \subseteq B_f$.

Let us suppose that condition 1 is true for all $b_i \in \{b_0, b_1, b_2, \cdots, b_k\}$, where $k \geq 1$, and conditions $2, 3$ are true for all $N \in \{1, 2, 3, 4, \cdots, b_k\} \backslash B_f$. We show that conditions $2, 3$ are true for all $N \in \{b_k + 1, b_k + 2, \cdots, b_{k+1} - 1\}$, and condition 1 is true for $b_i = b_{k+1}$.

Define $b_{\theta(k)}$ by $b_{k+1} = b_k + b_{\theta(k)}$, where $b_{\theta(k)} \in \{b_0, b_1, b_2, \cdots, b_k\}$.

In the following argument, we omit the first part when $b_{k+1} - b_k = 1$. So let us imagine that $b_{k+1} - b_k \geq 2$. We will prove that conditions $2, 3$ are true for $N \in \{b_k + 1, b_k + 2, \cdots, b_{k+1} - 1\}$ by proving this sequentially with N starting at $N = b_k + 1$ and ending at $N = b_{k+1} - 1$.

Note that once we prove condition 3 for any $N \in \{b_k + 1, \cdots, b_{k+1} - 1\}$, condition 2 will follow for this N as well. This is because if $N = b_k + (N - b_k)$, where $1 \leq N - b_k < b_k$, and $g(N) = g(N - b_k)$, then $g(N) = g(N - b_k) \leq N - b_k < N$. Note that $g(N - b_k) \leq N - b_k$ is always true. So let us now prove condition 3 is true for N as N varies sequentially over $b_k + 1, \cdots, b_{k+1} - 1$.

Since we are assuming that $b_{k+1} - b_k \geq 2$, this means that $f(1) < g'(b_k) = g(b_k)$ is assumed as well since $g'(1) = 1$. Therefore, $g(b_k + 1) = g(1) = 1$ is obvious. Therefore, suppose we have proved condition 3 for all $N \in \{b_k+1, b_k+2, \cdots, b_k+t-1\}$ where $b_k + t - 1 \leq b_{k+1} - 2$. This implies for all $N \in \{1, 2, 3, \cdots, b_k + t - 1\}, g(N) = g'(N)$. We now prove condition 3 for $N = b_k + t$. This means we know that $g(i) = g(b_k + i), i = 1, 2, 3, \cdots, t - 1$ and we wish to prove $g(t) = g(b_k + t)$. Recall that $g(t)$ is the smallest positive integer x such that the list

(1) $F(t-1, f(1)), F(t-2, f(2)), \cdots, F(t-x, f(x))$ contains exactly one 0 (which comes at the end of the list). Also, $g(b_k + t)$ is the smallest positive integer x such that the list

(2) $F(b_k+t-1, f(1)), F(b_k+t-2, f(2)), \cdots, F(b_k+t-x, f(x))$ contains exactly one 0 (which comes at the end). Since we are assuming that $g(i) = g(b_k+i), i = 1, 2, \cdots, t-1$, we know that the above two lists must be identical as long as $1 \leq x \leq t-1$. This follows from the definition of g since $g(N)$ tells us that $F(N, x) = 0$ when $1 \leq x \leq g(N) - 1$ and $f(N, x) = 1$ when $g(N) \leq x$. Now if $t \notin \{b_0 = 1, b_1, b_2 \cdots, b_{\theta(k)-1}\}$, we know from condition 2 that $g(t) < t$. This tells us that for list (1) the smallest x such that list (1) contains exactly one 0 satisfies $1 \leq x \leq t-1$. Therefore, since the two lists (1),(2) are identical when $1 \leq x \leq t-1$, this tells us that $g(t) = g(b_k + t)$.

Next, suppose $t \in \{b_0 = 1, b_1, b_2 \cdots, b_{\theta(k)-1}\}$. Of course, $g(t) = g'(t)$ since $t < b_k + t - 1$. Now if $g(t) = g'(t) < t$, the same argument used above holds to show that $g(t) = g(b_k + t)$. Now if $g(t) = g'(t) = t$, we know from the definition of how $b_{k+1} = b_k + b_{\theta(k)}$ is generated that $f(t) < g'(b_k) = g(b_k)$. Since $g'(t) = g(t) = t$, we know that the first $t-1$ members of each of the lists (1), (2) consists of all 1's since they are identical up this point and $g(t) = t$. Now in list (1), $F(t-t, f(t)) = F(0, f(t)) = 0$. Also, in list (2), $F(b_k+t-t, f(t)) = F(b_k, f(t)) = 0$ since $f(t) < g(b_k) = g'(b_k)$. This tells us that $g(t) = g(b_k + t) = t$ when $t \in \{b_0, b_1, \cdots, b_{\theta(k)-1}\}$, and $g(t) = g'(t) = t$.

Of course, $g(t) = g(b_k + t)$ is what we wished to show.

Finally, we show that $g(b_{k+1}) = g'(b_{k+1})$. Recall that $b_{k+1} = b_k + b_{\theta(k)}$ where $g'(b_{\theta(k)}) = g(b_{\theta(k)}) = b_{\theta(k)}$ and $f(b_{\theta(k)}) \geq g'(b_k) = g(b_k)$. We now know that $\forall N \in \{1, 2, 3, 4, \cdots, b_{k+1} - 1\}, g(N) = g'(N)$. Also, we know that $g(i) = g(b_k + i), i = 1, 2, 3, \cdots, b_{\theta(k)} - 1$. Since $g(b_{\theta(k)}) = b_{\theta(k)}$ we know that all terms in the following sequence are 1's except the final term which is 0.

(3) $F(b_{\theta(k)} - 1, f(1)), F(b_{\theta(k)} - 2, f(2)), F(b_{\theta(k)} - 3, f(3)), \cdots, F(1, f(b_{\theta(k)} - 1)),$ $F(0, f(b_{\theta(k)})) = 0$. Now $g(b_{k+1}) = g(b_k + b_{\theta(k)})$ is the <u>position</u> of the first 0 in the following sequence where we note that the <u>position</u> of a term $F(b_k + b_{\theta(k)} - i, f(i))$ is i.

(4) $F(b_k+b_{\theta(k)}-1, f(1)), F(b_k+b_{\theta(k)}-2, f(2)), \cdots, F(b_k+1, f(b_{\theta(k)}-1)), \underline{F(b_k, f(b_{\theta(k)}))},$ $F(b_k-1, f(b_{\theta(k)}+1)), \quad F(b_k-2, f(b_{\theta(k)}+2)), \cdots, F(1, f(b_{\theta(k)}+b_k-1)), F(0, f(b_{\theta(k)}+b_k)) = 0.$

Since $g(b_k + i) = g(i), i = 1, 2, \cdots, b_{\theta(k)} - 1$, we know that the first $b_{\theta(k)} - 1$ terms of (4) are 1's since the first $b_{\theta(k)} - 1$ terms of (3) are 1's. Now $F(b_k, f(b_{\theta(k)})) = 1$ from the definition of g since $f(b_{\theta(k)}) \geq g'(b_k) = g(b_k)$. Note that the last term in (4) is 0. Recall that for all $N \in \{1, 2, 3, \ldots, b_{k+1} - 1\}, \quad g(N) = g'(N)$.

From (4), we see that $g(b_{k+1})$ is the smallest $b_{\theta(k)} + \overline{x}, 1 \leq \overline{x} \leq b_k$ such that $F(b_k - \overline{x}, f(b_{\theta(k)} + \overline{x})) = 0$. Since $F(b_k - b_k, f(b_{\theta(k)} + b_k)) = 0$, we see that $g(b_{k+1})$ is the <u>smaller</u> of $b_{\theta(k)} + b_k = b_{k+1}$ and the smallest $b_{\theta(k)} + \overline{x}, 1 \leq \overline{x} < b_k$, such that $F(b_k - \overline{x}, f(b_{\theta(k)} + \overline{x})) = 0$ if such a $b_{\theta(k)} + \overline{x}$ exists. Now $F(b_k - \overline{x}, f(b_{\theta(k)} + \overline{x})) = 0,$

when $1 \leq \overline{x} < b_k$, if and only if $f(b_{\theta(k)} + \overline{x}) < g(b_k - \overline{x})$. Since $b_{\theta(k)} = b_{k+1} - b_k$ and since $g(b_k - \overline{x}) = g'(b_k - \overline{x}) = g'(\overline{g}(b_k - \overline{x}))$, if we compare the above definition of $g(b_{k+1})$ with the definition of $g'(b_{k+1})$ given earlier in this paper, we see that $g(b_{k+1}) = g'(b_{k+1})$. ■

Observation: We now explain why we defined B_f the way that we did. Assuming that B_f is infinite, we know from the definition of B_f that $b_{i+1} - b_i \leq b_i$.

Also, from the definition of b_{i+1}, we know that $g'(b_{i+1} - b_i) = b_{i+1} - b_i$.

Also, from the definition of $g'(b_{i+1})$ (since $\overline{x} \geq 1$ in the definition), we know that $g'(b_{i+1}) > b_{i+1} - b_i = g'(b_{i+1} - b_i)$. Thus $g(b_{i+1}) > g(b_{i+1} - b_i)$.

However, from statement 3 (at the beginning of the proof) we know that for all N satisfying $b_i < N < b_{i+1}$ it is true that $g(N) = g(N - b_i)$. This change at b_{i+1} is precisely why we defined B_f the way that we did.

The misere version: To win at the misere version (N, x) of dynamic nim, simply use the theory to win the game $(N - 1, x)$, so that your opponent is forced to take the last counter. The reader may like to figure out the strategy for the following variation. Suppose $S \subseteq Z^+ \cup \{0\}$, and the game is over as soon as $N \in S$, N being the pile size. In this paper $S = \{0\}$.

A difficult problem is to find (with proof) functions $f : Z^+ \to Z^+$ such that B_f and g' satisfy the following. $\{b_i : b_i \in B_f, g'(b_i) < b_i\}$ is infinite and $\{b_i : b_i \in B_f, g'(b_i) = b_i\}$ is infinite. We now give two examples of such functions. In both examples, B_f and g' will store the strategy extremely efficiently since the members of B_f grow exponentially.

Example 1: Define $f(n) = n, n \neq 8^k, k = 0, 1, 2, 3, \cdots, f(8^k) = 4 \cdot 8^k$.

Then $B_f = \{a \cdot 8^b : a, b$ integers, $1 \leq a \leq 7, 0 \leq b\}$, and $g'(a \cdot 8^b) = \phi(a) \cdot 8^b$, where $\phi(1) = 1, \phi(2) = 2, \phi(3) = 3, \phi(4) = 4, \phi(5) = 2, \phi(6) = 2, \phi(7) = 3$.

Noting that $g'(a \cdot 8^b) \leq 4 \cdot 8^b$, we leave the proof as an exercise for the reader. The proof is by induction in blocks of seven starting with proving that $\{1, 2, 3, 4, 5, 6, 7\} \subseteq B_f$ and $g'(1) = 1, g'(2) = 2, g'(3) = 3, g'(4) = 4, g'(5) = 2, g'(6) = 2, g'(7) = 3$.

Example 2: Define $f(n) = n$, if n is even and $f(n) = 4n$, if n is odd.

Instead of calling $B_f = (b_0 = 1 < b_1 < b_2 < \cdots)$, it is more convenient to call $B_f = (a_1 = 1 < b_1 < c_1 < a_2 < b_2 < c_2 < d_2 < a_3 < b_3 < c_3 < d_3 < \cdots < a_i < b_i < c_i < d_i < \cdots)$. The first few terms of B_f and the corresponding g' are computed by the following recursion. First, we define a strictly increasing sequence $\Delta_2, \Delta_3, \ldots$ recursively as follows: $\Delta_2 = 1, \Delta_3 = 3$ and for all $i \geq 4, \Delta_i = \Delta_{i-1} + 4\Delta_{i-2}$. Also, define $a_1 = 1, g'(a_1) = 1, b_1 = 2, g'(b_1) = 2, c_1 = 3, g'(c_1) = 3, a_2 = 4, g'(a_2) = 4, b_2 = 5, g'(b_2) = 2, c_2 = 6, g'(c_2) = 2, d_2 = 7, g'(d_2) = 7$. For all $i \geq 3$, define $a_i, g'(a_i), b_i, g'(b_i), c_i, g'(c_i), d_i, g'(d_i)$ recursively as follows: $a_i = d_{i-1} + \Delta_i, g'(a_i) = a_i, b_i = a_i + \Delta_i, g'(b_i) = 2\Delta_i, c_i = b_i + \Delta_i, g'(c_i) = 2\Delta_i, d_i = c_i + \Delta_i, g'(d_i) = d_i$.

In the third example, we give an example of a function f such that $B_f = Z^+$. If $B_f = Z^+$, it is easy to prove that g must be bounded. The reader can show that if either g or f is bounded, then eventually g must become periodic.

Example 3: Let $f(1) = 4, f(n) = 2, n \geq 2$. Then $g(1) = 1, g(2 + 4k) = 2$,

$g(3+4k)=3, g(4+4k)=4, g(5+4k)=2, k=0,1,2,3,\cdots$. Also, it is easy to see that $B_f = Z^+$. Of course, $g' = g$ on B_f.

Appendix. The functions f, g, g' and the base B_f described in this paper have a great many properties, a few of which are listed below. Recall that f is given, and it determines the other three.

1. g is periodic on Z^+ if and only if B_f is finite.

2. If f is bounded, there exists a positive integer a such that g is periodic on $[a, \infty)$.

3. If g is bounded, then there exists a positive integer a such that g is periodic on $[a, \infty)$.

4. Suppose the positive integer a satisfies $f(i) < g(a)$ for all $i = 1, 2, 3, \ldots, a$. Then g is periodic on Z^+ and $[1, a]$ is a period.

5. g is bounded if and only if $\{b_i : \quad b_i \in B_f, g(b_i) = b_i\}$ is finite.

6. There exists a positive integer a such that g is periodic on $[a, \infty)$ if and only if g is bounded on B_f.

References

[1] Elwyn Berlekamp, John H. Conway, and Richard Guy, *Winning Ways*, Academic Press, Vol. 1, A. K. Peters, 2001.

[2] John H. Conway, *On Games and Numbers*, A. K. Peters, 2001.

[3] Epp, Richard and Thomas Ferguson, "A Note on Takeaway Games", *The Fibonacci Quarterly*, 18:4 (1980):300-303.

[4] Richard K. Guy, Fair Game, 2nd. ed., COMAP, New York, 1989.

[5] Holshouser, A., J. Rudzinski, and H. Reiter, Dynamic One-Pile Nim, submitted.

[6] A. J. Schwenk, "Take-Away Games", *The Fibonacci Quarterly*, 8:3 (1970):225-234.

ams classification: 91A46