1 Introduction

The mathematical subject of 'game theory' is less than a hundred years old, and has been applied broadly, especially in economics, to analyze a huge variety of real-life situations. During the past 80 years, two major branches of game theory have emerged, classical games and combinatorial games.

A two-player classical game is a game played with a *payoff matrix* between a *row player* and a *column player*. Each one chooses a strategy, unbeknownst to the other. The payoff (usually written with respect to the row player) is the entry in the row and column selected. When the game is played multiple times, as is often the case in real life, the players must select a probability distribution on their strategy spaces, and this leads to an expected payoff, which each player attempts to maximize. Linear programming is the typical mathematical tool used to find these best strategies. Classical game theory is used to analyze games which may involve chance, imperfect information (not knowing your opponent's move), simultaneous moves, and/or the possibilities for cooperation. For example, poker, Monopoly, and the global nuclear arms race are not combinatorial games. See `http://en.wikipedia.org/wiki/Minimax`

Mathematical competitions tend to focus on combinatorial games, as we will here. A combinatorial game is a set of players, a set of positions, a set of allowable moves, and a set of payoffs for various outcomes. In combinatorial games, there is no randomness, players take turns, and the set of allowable moves is public information, as is each player's move.

A *combinatorial game* is a set of players, a set of positions, a set of allowable moves, and a set of payoffs for various outcomes. In combinatorial games, there is no randomness, players take turns, and the set of allowable moves is public information, as is each player's move. Combinatorial games can be *impartial*, meaning that the set of moves allowed to each player is the same, or *partisan*, meaning that the players' sets of allowable moves may be different. Single and multiple pile nim games and Go are impartial, while chess and Tic-Tac-Toe are partisan, because the player with the white pieces (or $\times$'s) cannot move any black piece. Nim is a family of games in which two players begin with a pile or piles of counters or stones. Players alternate removing some number of stones from the pile or piles. The winning player is the last one to make an allowable move. Different versions of Nim result from different rules about how many stones can be removed at each step, and different starting positions (numbers of stones and number of piles).

Typically the aim of a combinatorial game problem is to identify a winning strategy, either for the starting player or for the opponent. We say player A has a winning strategy if, whatever moves B makes, there is some set of moves that A can make so that A is guaranteed to win.

In the 1930s, it was shown (the Sprague-Grundy theorem) that every impartial combinatorial game is equivalent to some variant of nim, which has been completely solved. Thus, any game in this category can, in theory, be 'worked out', though the actual computations may be far beyond current computational power. So, one approach to solving a combinatorial game is to translate it into a variant of nim. We will see an example of this later in this paper. However, it may be difficult to translate a game into nim, and solving the nim variant may also be difficult, so this is far from a universal strategy.

This paper is divided into nine sections. In the second section, we present two algorithms for expressing a given positive integer n as a base-b numeral. We use a base of 6 for the examples. In the third section, we present two algorithms for finding the base-b representation of a fraction. In section 4, we discuss three unusual place value type representations: Fibonacci, Factorial, and Balance-pan representations, and their arithmetics. In section 5, we discuss the Prime Factorization method of representation and provide a table of values for all five methods of representation in the paper. In section 6, we discuss the relationship between integer representation and game playing strategies for *static* one-pile nim. In particular, we show how finding the right representation often leads to a convenient method for finding a winning strategy in counter pickup games. In section 7 we present an introduction to *dynamic* counter pickup games. These are games for which the number of counters that can be removed changes according to some game parameters as the game is in progress. In section 8, the reader will find a collection of exercises and problems related to representation of integers and fractions. Section 9 contains problems on combinatorial games. Readers whose main interest is combinatorial games can skip all of the first four sections except the repeated subtraction method for representing integers and the Fibonacci representation.

2 Place value and base 6 representation

The *place value* interpretation of 4273 is $4000 + 200 + 70 + 3$, which is a *sum* of *multiples* of *powers* of 10. The relevant powers of 10, $10^3 = 1000, 10^2 = 100, 10^1 = 10$, and $10^0 = 1$ all have *coefficients* or multipliers, $4, 2, 7$, and 3. Thus $4 \cdot 10^3, 2 \cdot 10^2, 7 \cdot 10^1$, and $3 \cdot 10^0$ are **multiples** of **powers** of 10 and therefore 4273 is a **sum** of **multiples** of **powers** of 10. Notice that the digit 3 is the remainder when 4273 is divided by 10. We will see below that the division process enables us to express a given integer as the sum of multiples of powers of b, where b is a positive integer bigger than 1.

For convenience, let us assume for sections 2 and 3 that $b = 6$. The same procedures work no matter what the value of b is, but fixing the value of b here makes discussion much easier. The notation 2113_6 is interpreted as a **sum** of **multiples** of **powers** of 6, just as the 4273 was above. The *subscript* 6 must be attached unless we are using base 10, because 10 is the default value of the base. Thus $2113_6 = 2 \cdot 6^3 + 1 \cdot 6^2 + 1 \cdot 6^1 + 3 \cdot 6^0 = 477$. The process of finding the decimal (ie, base 10) value of a number from its base 6 representation is called *interpreting.* Thus we interpreted 2113_6 as 477. The reverse process, that of finding the base 6 representation of an integer expressed in decimal notation is harder and more interesting. There are two methods, (a) *repeated subtraction* and (b) *repeated division.* Each method has some advantages over the other.

To see how to use repeated subtraction, first make a list of all the integer powers of 6 that are not bigger than the number we are given. In the case of 477, we need the powers $6^0 = 1, 6^1 = 6, 6^2 = 36$, and $6^3 = 216$. Next repeatedly subtract the largest power of 6 that is less than or equal to the *current number* (which changes during the process). So we have $477 = 216 + 261$. At this point our current number becomes 261 and we repeat the process. Then $477 = 216 + 261 = 216 + 216 + 45$, and our current number is 45. Repeating the process on 45 gives $45 = 36 + 9$ and incorporating that in the above gives $477 = 2 \cdot 216 + 45 = 2 \cdot 216 + 1 \cdot 36 + 9$. Continuing this with 9 leads to $477 = 2 \cdot 6^3 + 1 \cdot 6^2 + 1 \cdot 6^1 + 3 \cdot 6^0$, which is a sum of multiples of powers of 6, just what we want. Thus $477 = 2113_6$, just as we saw above. Repeated subtraction has two advantages over the repeated division method. First, it is closely related to the definition, hence it leads to a better conceptualization. Second, it can be used in other situations when repeated division cannot, as in the case of Fibonacci representation.

The repeated division method requires that we repeatedly divide the given integer by base 6 and record the remainder at each stage. First we divide 477 by 6 to get $477 \div 6 = 79.5$. We can interpret this as $477 = 6 \cdot 79 + 3$, so the *quotient* is 79 and the *remainder* is 3. Notice that the remainder can never exceed 5 since in such a case the quotient would have been larger. Next divide the quotient by 6 and record the new quotient and the remainder. Thus $79 = 6 \cdot 13 + 1$. Repeat the process $13 = 6 \cdot 2 + 1$ and finally, $2 = 6 \cdot 0 + 2$. Next write the remainders in reverse order, $2, 1, 1$, and 3 to get 2113_6 as the base 6 representation of 477. You'll see why the order must be reversed in the following example.

Example 1. Repeated Division To see why $477 = 2113_6$, we can repeatedly replace each quotient with its value obtained during the division process. Thus

$$\begin{aligned}
477 &= 6 \cdot 79 + 3 \\
&= 6(6 \cdot 13 + 1) + 3 \\
&= 6(6(6 \cdot 2 + 1) + 1) + 3 \\
&= 6(6 \cdot 6 \cdot 2 + 6 \cdot 1 + 1) + 3 \\
&= 6 \cdot 6 \cdot 6 \cdot 2 + 6 \cdot 6 \cdot 1 + 6 \cdot 1 + 3 \\
&= 2 \cdot 6^3 + 1 \cdot 6^2 + 1 \cdot 6^1 + 3 \cdot 6^0 \\
&= 2113_6
\end{aligned}$$

The advantage of repeated division is that it is computationally more efficient. Also, the method of justification can be applied in other situations (synthetic division and Euclidean algorithm).

3 Repeated Multiplication and Repeated Subtraction

In section 2 we saw two methods (algorithms) for writing a given integer in a base different from 10. Before we consider representing fractions, let's review the place value ideas in decimal notation. For example, 5.234 is, as in the first part, a sum of multiples of powers of 10. This time, the powers have (except one) negative exponents:

$$5.234 = 5 \cdot 10^0 + 2 \cdot 10^{-1} + 3 \cdot 10^{-2} + 4 \cdot 10^{-3}.$$

Using this interpretation as a guide, we can interpret 0.124_6 similarly, as a sum of multiples of (negative) powers of 6. Thus

$$\begin{aligned} 0.124_6 &= 1\cdot 6^{-1} + 2\cdot 6^{-2} + 4\cdot 6^{-3} \\ &= \frac{1}{6} + \frac{2}{36} + \frac{4}{216} = \frac{36+12+4}{216} \\ &= \frac{52}{216} = \frac{13}{54} \end{aligned}$$

As in the discussion of integers, there are two methods for dealing with numbers in the range $0 < x < 1$. They are called (a) *repeated subtraction* and (b) *repeated multiplication.* As before, each has advantages over the other.

Example 2. Repeated Subtraction To use the method of repeated subtraction on 13/54, first list the powers of 6 with negative integer exponents:

$$6^{-1} = 1/6,\ 6^{-2} = 1/36,\ 6^{-3} = 1/216, \ldots.$$

Find the largest of these powers of 6 and subtract it from the original number. Thus $13/54 - 1/6 = 13/54 - 9/54 = 4/54 = 2/27$. Therefore, $13/54 = 1/6 + 2/27$. Now repeat the process on the number 2/27. Note that $1/36 = 3/108$. Thus, $2/27 - 1/36 = 8/108 - 3/108 = 5/108$. Therefore, $2/27 = 1/36 + 5/108$. Putting this together with the arithmetic above, we have

$$\frac{13}{54} = \frac{1}{6} + \frac{1}{36} + \frac{1}{36} + \frac{1}{54}.$$

Again dealing with the extra part, $1/54 - 1/216 = 4/216 - 1/216 = 3/216$. At this point we can anticipate the final arithmetic:

$$\begin{aligned} \frac{13}{54} &= \frac{1}{6} + \frac{1}{36} + \frac{1}{36} + \frac{1}{216} + \frac{1}{216} + \frac{1}{216} + \frac{1}{216} \\ &= 1\cdot 6^{-1} + 2\cdot 6^{-2} + 4\cdot 6^{-3} \\ &= 0.124_6 \end{aligned}$$

The method of repeated multiplication is much quicker and does not require so much fraction arithmetic.

Example 3. Repeated Multiplication To find the base 6 representation of 13/54, we repeatedly multiply by 6. Following each multiplication by 6, split the result into its integer part and its fractional part:

$$\frac{13}{54} \cdot 6 = \frac{13 \cdot 6}{6 \cdot 9} = \frac{13}{9} = 1 + \frac{4}{9}.$$

Each integer part is a digit in the representation. Thus $13/54 = 0.1\ldots_6$. Now repeat the process using the new fractional part, 4/9:

$$\frac{4}{9} \cdot 6 = \frac{24}{9} = 2 + \frac{6}{9} = 2 + \frac{2}{3}.$$

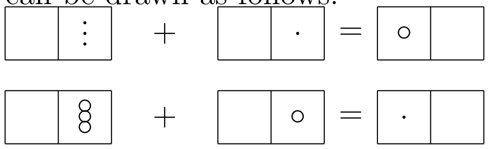
Thus $13/54 = 0.12\ldots_6$. Repeating the process, $\frac{2}{3} \cdot 6 = 4 + 0$. Since the fractional part is 0, we are done (why?). Thus, $\frac{13}{54} = 0.124_6$.

Of course, not all rational numbers have base 6 representations that terminate (ie, end in all 0's from some point on). But there is an easy way to tell, and a great notation to use when the representation does not terminate. Consider the problem of finding the binary representation of $\frac{1}{3}$. Using repeated multiplication, we get $\frac{1}{3} \cdot 2 = 0 + \frac{2}{3}$. Then $\frac{2}{3} \cdot 2 = 1 + \frac{1}{3}$. Thus we see the same fractional part $\frac{1}{3}$ occur again. The first two digits are 0 and 1, so we have $\frac{1}{3} = 0.01\ldots_2$, but we can see that the block 01 continues to recur. The slick way to write this number $0.01010101\ldots$ is $0.\overline{01}_2$. When the representation repeats in blocks, the number can be regarded as the sum of an infinite geometric series. In this case it is $2^{-2}+2^{-4}+2^{-6}+\cdots$. There is a formula for finding the sum of the geometric series $a+ar+ar^2+ar^3+\cdots$. It is $\frac{a}{1-r}$, and this holds whenever $|r| < 1$. Thus $2^{-2} + 2^{-4} + 2^{-6} + \cdots = \frac{2^{-2}}{1-2^{-2}} = \frac{1/4}{3/4} = \frac{1}{3}$, just as we knew.

4 Place Value with Negative Bases

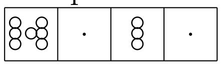
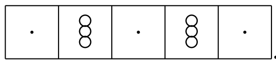
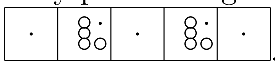
In this section we study the consequences of using a negative base for arithmetic. As we did in the previous section, we'll pick a sample base and stick with it throughout. You'll see easily how to modify the ideas when other bases are used. We'll pick negative 4 as our base. Here we allow ourselves the digits $0, 1, 2, 3$. Let us first **interpret** a number written in base -4. For example take 113.3_{-4}. We interpret this as a sum of multiples of powers of -4: $1 \cdot (-4)^2 + 1 \cdot (-4)^1 + 3 \cdot (-4)^0 + 3 \cdot (-4)^{-1} = 16 - 4 + 2 - 3/4 = 13.25$. Thus, we write $13.25 = 113.3_{-4}$. The methods for finding the base negative four representation of a positive integer are interesting. Also of interest are methods for finding the base -4 representation of rational numbers r satisfying $0 < r < 1$. We can find the base -4 representation of 13.25 by combining these two methods.

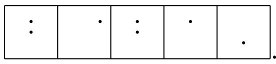
Incidentally, the paper on Exploding Dots, Antidots and Black Holes is relevant here. The machine we can use is denoted $\boxed{-1 \leftarrow 4}$. Its called this because when four dots accumulate in a box, they explode, causing an antidot to be formed in the next box to the left. Similarly when four antidots accumulate, they explode to give a dot in the box to the left. This machine can be drawn as follows:


and

Example 1. Repeated Division To see why $477 = 21211_{-4}$, we can repeatedly replace each quotient with its value obtained during the division process. Its important to remember that the remainders cannot be negative numbers. Thus

$$\begin{aligned}
477 &= -4 \cdot -119 + 1 \\
&= -4(-4 \cdot 30 + 1) + 1 \\
&= -4(-4(-4 \cdot -7 + 2) + 1) + 1 \\
&= -4(-4(-4(-4 \cdot 2) + 1 + 1) + 1 \\
&= -4(-4(-4(-4 \cdot 2 + 1) + 2) + 1) + 1 \\
&= 2(-4)^4 + 1(-4)^3 + 2(-4)^2 + (1(-4)^1 + 1(-4)^0 \\
&= 21211_{-4}
\end{aligned}$$

Here's how the exploding dot machine above would process the number 477. First, there would be 119 explosions that would produce 119 antidots in the second box and one dot in the right box. Then 29 explosions would take place, producing 29 dots in the third box and 3 antidots in the second box. Then 7 explosions would take place to produce 7 antidots in the fourth box with one dot left in the third box:  The final explosion produces: . So, can we say that the base -4 representation of 477 is $1-31-31$? Of course not. We can use only positive digits. So what can we do? Try adding some dot-antidot pairs. .

From here its easy: . In other words, 21211_{-4}.

The algorithms for finding the base -4 representation of fractions is even more interesting. My AwesomeMath student Eliot Levmore, suggested the following algorithm, related to repeated multiplication. To find the base -4 representation of $7/20$, first note that our number is positive, so it looks like $1.abcd....$ That means the $.abcd...$ has value $7/20 - 1 = -13/20$. Multiply $-13/20$ by -4 to get $52/20 = 13/5 = 3-2/5$, so the digit a is 3. Then multiply $-2/5$ by -4 to get $8/5$ which we can write as $2 - 2/5$. Our representation is $1.32cd....$ Now $-2/5 \cdot -4 = 8/5$ again, and we can see that the digit 2 repeats. Thus $7/20 = 1.3\overline{2}_{-4}$. Can you prove that this is correct? Which rational numbers less than 1 require a digit 1 in the unit's position? Levmore again provides the answer. To see what it is, ask yourself the question, What is the largest rational number representable as $0.x_1x_2\ldots$?

In algebra, you learn a method for converting a repeating decimal to a ratio of two integers. We can do that here also. Let $t = 1.3\overline{2}_{-4}$. Then $16t = 132.\overline{2}$ and $16t-t = 15t = 132.2_{-4} - 1.3_{-4} = 132.3 = 21/4 \div 15 = 7/20$. This algorithm is not perfect, however because the subtraction idea can lead to digits larger than 3. Can you devise another method that avoids this problem?

Here's an idea. To find the base -4 representation of a fraction, we repeatedly multiply by 16, and produce two digits at a time. This way we can avoid the difficulties posed by the negative numbers.

5 Fibonacci, Factorial, and Balance-Pan Enumeration

Fibonacci Representation The Fibonacci numbers $F_1 = 1, F_2 = 2, F_3 = 3, F_4 = 5\ldots$ are defined so that after the first two, every one is the sum of the last two. In other words, $F_1 = 1, F_2 = 2$ and $F_{n+2} = F_n + F_{n+1}$. Thus the sequence is $1, 2, 3, 5, 8, 13, 21, 34, 55, 89\ldots.$ In the case of Fibonacci representation, we need only two digits, 0 and 1. These represent the absence or presence of the corresponding Fibonacci number. To represent a number in Fibonacci representation, use the method of repeated subtraction.

Example 4. Fibonacci Representation To find the Fibonacci representation of 100, find the largest Fibonacci number less than or equal to 100. Then subtract it and repeat the process. Thus $100 = 89 + 11$. Thus

$100 = 89 + 11 = 89 + 8 + 3 = 1000010100_f$. Of course, the 1's tell us which Fibonacci numbers are added, and the 0's tell us to leave out the number: 1000010100_f means $1F_{10}+0F_9+0F_8+0F_7+0F_6+1F_5+0F_4+1F_3+0F_2+0F_1$. Notice that the representation 1000010100_f has at least one 0 between each pair of 1's. Try to figure out why this is always the case this before reading on. We'll return to this representation later. How can we do arithmetic with numbers represented this way? Addition is not very hard. Let's use the addition $87 + 31$.

Example 5. Fibonacci Arithmetic In the notation we (slightly) abuse the notation by using the coefficient 2 at times.

	89	55	34	21	13	8	5	3	2	1
87		1	0	1	0	1	0	1	0	0
+31				1	0	1	0	0	1	0
		1	0	2	0	2	0	1	1	0
		1	0	2	0	2	1	0	0	0
		1	0	2	1	1	0	0	0	0
		1	1	1	0	1	0	0	0	0
118	1	0	0	1	0	1	0	0	0	0

The addition process repeatedly makes use of the fact that the sum of two successive Fibonacci numbers is the next one. In the representation, therefore, you never need to have two successive 1's. Another example might be helpful here. How would you carry out $21 + 21$? That is $1000000_f + 1000000_f = 2000000_f = 1110000_f = 10010000_f = 34 + 8 = 42$ Can you devise an algorithm for multiplication?

Factorial Representation Here the idea is to represent each number as a sum of multiples of factorials. The basic building blocks are the numbers $1 = 1!, 2 = 2!, 6 = 3!, 24, 120, 720, \ldots$. The coefficients allowed for $n!$ are the numbers from 1 up to n. Of course, using $n + 1$ as a coefficient for $n!$ would not be needed since $(n + 1)n! = (n + 1)!$. The table below lists the representations of the first twelve positive integers. How can we find the factorial base for a positive integer N? The answer is by repeated division. First, divide by two and write the remainder in the rightmost position. Of course the remainder is either 1 or 0 depending on the parity of the N. Next divide the quotient by three, and again write down the remainder. Continue this process until the quotient is zero.

Example 6. Repeated Division To find the Factorial representation of $N = 127$ first divide by 2. The first remainder is 1, and the quotient is 63.

Dividing 63 by three yields a quotient of 21 and a remainder of 0. Then dividing 21 by four yields quotient 5 with remainder 1. Finally, divide by five to get a quotient of 1 and a remainder of 0. Thus $127 = 10101_!$. That is, $127 = 5! + 3! + 1!$.

Arithmetic in factorial notation is not very hard. Let's pursue addition.

Example 7. Factorial Arithmetic Consider the problem $65 + 21$, which in factorial notation is $2221_! + 311_!$ since $65 = 2 \cdot 4! + 2 \cdot 3! + 2 \cdot 2! + 1 \cdot 1!$ while $21 = 3 \cdot 3! + 1 \cdot 2! + 1 \cdot 1!$. So

$$\begin{array}{r} 2 \cdot 4! + 2 \cdot 3! + 2 \cdot 2! + 1 \cdot 1! \\ + 3 \cdot 3! + 1 \cdot 2! + 1 \cdot 1! \\ \hline 2 \cdot 4! + 5 \cdot 3! + 3 \cdot 2! + 2 \cdot 1! \end{array}$$

But $5 \cdot 3! = (4 + 1) \cdot 3! = 4 \cdot 3! + 3! = 4! + 3!$. So the sum is just $3 \cdot 4! + 2 \cdot 3! + 2! = 3210_!$.

Balance-Pan Enumeration Now let's turn our attention to balance-pan enumeration. Think about how you would arrange four weights on a two-pan balance to weigh out each of the numbers from 1 to 40, and what weights would you use for such a project. Since there are three things you can do with each weight (put it on the left pan, the right pan, or not use it), there are at most $3^4 = 81$ arrangements of the four weights. One of these is to do nothing with each weight, and for every other arrangement, there is the opposite arrangement where each weight on the left pan is moved to the right pan, and vice-versa. So there are just 40 possible values to be weighted with four weights. A little playing with this leads to the possibility of using the first four powers of 3, $3^0 = 1, 3^1 = 3, 3^2 = 9$, and $3^4 = 27$. We can let the digits $0, 1$, and $\overline{1}$ mean a) don't use the weight, b) put the weight on the left pan, and c) put the weight on the right pan. We'll always put the object to be weighed in the right pan, so we will need to arrange the weights so the sum of the weights in the left pan is at least as large as the sum of the weights in the right pan. Now the representation $11\overline{1}0_b = 1{\cdot}3^3+1{\cdot}3^2-1{\cdot}3^1+0{\cdot}3^0 = 27+9-3 = 33$, for example.

Example 8. Balance Pan Arithmetic

$$\begin{array}{r} 1\ \overline{1}\ 1\ 1 \\ \underline{\times\ 1\ 0\ \overline{1}} \\ \overline{1}\ 1\ \overline{1}\ \overline{1} \\ \underline{1\ \overline{1}\ 1\ 1\ 0\ 0} \\ 1\ \overline{1}\ 1\ \overline{1}\ \overline{1}\ \overline{1} \end{array}$$

Of course, this is just a slick way to show that $22 \times 8 = 176$.

To do arithmetic of integers represented in balance-pan representation, we need the addition and multiplication tables for digits, just as we learned in third grade for decimal representation.

$+$	$\overline{1}$	0	1
$\overline{1}$	$\overline{1}1$	$\overline{1}$	0
0	$\overline{1}$	0	1
1	0	1	$1\overline{1}$

$\times$	$\overline{1}$	0	1
$\overline{1}$	1	0	$\overline{1}$
0	0	0	0
1	$\overline{1}$	0	1

Of course, the balance pan representation of a number is closely related to its ternary representation. In fact $M = (a_k a_{k-1} \dots a_0)_b$ where $a_i \in \{0, \overline{1}, 1\}$ if and only if $M = \sum_{i \in P} 3^i - \sum_{i \in N} 3^i$, where P is the set of indices i for which $a_i = 1$ and N is the set of indices i for which $a_i = \overline{1}$. For yet another example, consider the case $M = 15 = 3^3 - 3^2 - 3$. Then $a_3 = 1, a_2 = \overline{1}, a_1 = \overline{1}$, and $a_0 = 0$.

6 Prime Notation

Here is perhaps the most interesting of the four methods of enumeration. We are all well aware of the uniqueness of prime factorization of positive integers. If we agree to write all the primes in every factorization, making use of the fact that $p^0 = 1$, we get a representation of positive integers. $1 = 2^0, 2 = 2^1, 3 = 3^1 2^0, 4 = 2^2, 5 = 5^1 3^0 2^0$, and $6 = 3^1 2^1$. Now write the list of exponents in the same order as above: $1 = 0_p, 2 = 1_p, 3 = 10_p, 4 = 2_p, 5 = 100_p$, and $6 = 11_p$. In this system, multiplication is especially easy. For example, $12101_p \times 21001_p = 33102_p$. Can you figure how we did this?

7 Static one-pile nim

The point of this section is to provide a nearly trivial example that illustrates the ideas we encounter in the next section, and also to develop some notation. Static one-pile nim is a counter pickup game contested by two players. The initial number of counters, denoted by n, can change from game to game. The maximum number of counters k that can be removed on each turn is fixed throughout the game. The notation $N_4\,(20)$ means that there are initially $n = 20$ counters and that up to $k = 4$ counters can be removed on each turn. The winner, as in all the games we discuss here, is the last player to make a move (i.e., the player who takes the last counter). The games $N_k\,(n)$ of static one pile nim are completely understood. They are among the simplest of combinatorial games. *Chess*, *checkers*, *tic-tac-toe*, and *go* are more complicated examples. The 'solution' of $N_4\,(20)$ is obtained by first noting that there are 21 "positions" in the game, represented by the integers $0, 1, 2, \cdots, 20$. We call 20 the *initial position* and 0 the *terminal position* of $N_4\,(20)$. The sequence of positions $20 \mapsto 18 \mapsto 15 \mapsto 14 \mapsto 10 \mapsto 9 \mapsto 5 \mapsto 2 \mapsto 0$, such that for each $x \mapsto y$ a pile of size y is obtainable from a pile of size x, is called a 'play' of the game. Notice that there are eight moves in this play of the game. Since eight is an even number, the second player wins this game. In fact any such game with an even number of moves is won by the second player, and those plays of the game with an odd number of moves are won by the first player. Thus, the issue is whether the first player can play in such a way as to force the game to have an odd number of moves. Alternatively, the second player would like to force the game to end after an even number of moves.

One approach to solving counter pickup games like these is to find a handy representation for the pile sizes and an easily understood method for finding optimal moves. The following example illustrates the idea.

Example 9. Consider the game $N_4\,(20)$ again. Notice that each number $l = 0, 1, 2, \cdots, 20$ can be represented in the form $l = 5t + u$ where l is the size of the pile and $0 \le t \le 4$ and $0 \le u \le 4$. With this in mind, for example, we could write $17 = 5 \cdot t + u = 5 \cdot 3 + 2$, or 32(three 5's and two 1's) or $111 :$. (The final dot is a 'period'.) The numbers 5 through 13 can be written using this notation as follows:
$5 = 1$ (1 summand); $6 = 1\cdot$ (2 summands); $7 = 1 :$ (2 summands); $8 = 1\vdots$ (2 summands); $9 = 1 ::$ (2 summands); $10 = 11$ (2 summands); $11 = 11\cdot$ (3 summands); $12 = 11 :$ (3 summands) and $13 = 11\vdots$ (3 summands). Each 1(representing 5) is a summand and each **set** of dots $\cdot, :, \vdots, ::$ is a summand.

With this understanding, it is easy to see that a move from the position $20 = 1111$ **cannot reduce** the number of summands. This is true because each of the representations for $19, 18, 17$ and 16 has four summands, just as 20 does. Denote the set of relevant multiples of five by S. Thus $S = \{0, 5, 10, 15, 20\}$. Let U denote the rest of the attainable positions. Now it is easy to see that each move **from** a position in S results in a position in U. On the other hand, from each position p in U there is **some move** that reduces the number of summands, for example: $111 :\mapsto 111$. In other words, a player confronted with a pile size of 5, 10, 15, or 20 cannot reduce the number of summands, but a player confronted with a pile size that is not a multiple of 5 (all the other positions) **can** reduce the number of summands. Delightfully, the positions (pile sizes) are divided into two subsets S and U (called *safe* and *unsafe*) so that the following diagram describes the moves. Here we are going to use the notation O for oasis position instead of safe and poison position P instead of unsafe U.

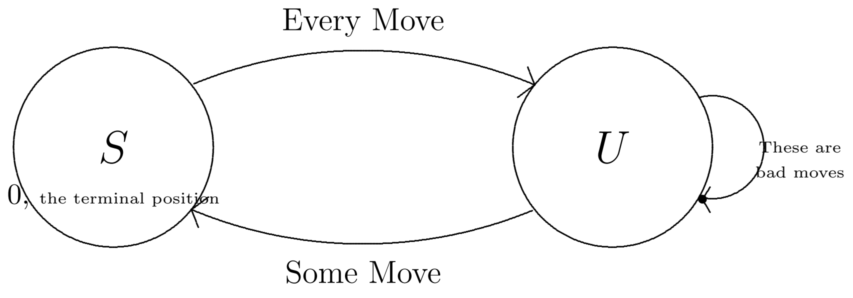


Fig. 1

In the game $N_4(20), S = \{0, 5, 10, 15, 20\}$ and $U = \{1, 2, 3, 4, 6, 7, \cdots\}$. Note that each move from U to S **reduces** the number of summands in the representation, whereas none of the moves from positions of S reduce the number of summands. Notice also that if a player is able to make such a summand-reducing move, he can continue to make such moves on succeeding turns.

The S vs U classification of the positions is the classic method for solving combinatorial games. Our main contribution is to note that in certain dynamic games, there is a representation of the pile size (we call these g-base representations) for which one of the players can repeatedly reduce/not increase the number of summands while the opposing player cannot reduce/must increase the number of summands.

8 Dynamic Nim

We finally come to the application part of the handout. Two players are engaged in a single pile game of Nim in which the number of counters that can be removed varies during the play of the game. The rules for this game specify a function f, called the move function, that is used to determine the maximum number of counters that can be removed as a function of the number removed on the previous move. The winner is the last player to make a move. Dynamic one-pile nim with move function f and staring pile size n is denoted $N_f(n)$.

We focus here on the two games $N_i(n)$ and $N_d(n)$ where i is the identity function and d is the doubling function. The *positions* in the game are ordered pairs (t, k) where t is the pile size and k is the maximum number of counters that can be removed on the next turn. A *move* is an ordered pair of positions $(t, k) \mapsto (t - r, f(r))$, where $r \leq k$. The position $(t - r, f(r))$ results from taking r counters from the pile of t counters.

Let's play $N_i(100)$ first. So make 100 marks on your paper, decide which player moves first and let them choose some number of markers less than 100. The second player can then take only as many as was taken on the previous play. Suppose the first player takes 1 counter. Then the rest of the moves are completely determined, and the second player will continue to move to the positions with an even number of counters, including (eventually) the zero position. So the first player must not take 1. Suppose he takes an odd number of counters. Then the second player can take 1 counter, and we are back to the same situation as above, a win for the second player. Thus, the first player must take an even number of counters and, by the same reasoning, each move after that should also be an even number of counters. Hence, we may as well assume that the counters are glued together in pairs, and that we start with only 50 pairs. Applying the same reasoning, the first player cannot take just 1 of these pairs. In fact he must take an even number, and so must the second player. Hence we may as well be playing the game $N_i(25)$ with counters that are glued together in bunches of four. Now there is a move for the first player that clearly wins: he must take one bunch of four. This translates into the move $(100, 99) \mapsto (96, i(4)) = (96, 4)$. The number 99 in the move above is arbitrary. It simply means that player 1 cannot win on the first move. This gluing ideas brings to mind the possibility of using binary notation. Suppose we're playing $N_i(100)$ and the first player can take up to 10 counters. Then we denote the first position $(100, 10)$. Write 100 in binary form to get $2^6 + 2^5 + 2^2 = 1100100_2$ (three summands). Thus one move the first player can make is to $(1100000_2, 4)$, by taking 4 counters, reducing the number of summands from three to two. Note that the next player cannot reduce the number of summands. In fact, the next move results in $95, 94, 93,$ or 92 counters, and all these have binary representations with at least five summands, the least value of which is either 1 or 2. Thus, the first player can again reduce the number of summands, and following that, the second player cannot. We can use the following theorem to play this game. An easy-to-remember rule for winning $N_i(n)$ is to remove the largest power of 2 that divides the pile size.

Theorem 1. *Consider the game $N_i(n)$ where $n = e_l e_{l-1} \cdots e_0 = \sum_{j=0}^{j=l} e_j \cdot 2^j$ be the binary representation of the pile size n. Then the first player has a winning move from the position (n, k) if and only if $2^t \leq k$ where $e_t = 1$ and for all $j < t$, $e_j = 0$. In other words, if the player can reduce the number of summands in the binary representation of n, then he can win. If not, he can't win.*

The second example $N_d\,(100; 10)$ is a game we call Fibonacci Nim. The subscript d in this case means *doubling*, the 100 is the number of counters at the start and 10 is the maximum size of the first move. This dynamic one pile nim is of the type (a) where the size of a move is determined by the size of the previous move. Here we allow a player to take as many as twice the number taken on the previous move, thus the "doubling function". The best way to denote a position in this game is as an ordered pair (n, k) where n is the size of pile and k is the maximum number of counters that can be removed on the next turn. Thus $(100, 10) \mapsto (92, 16)$ means that the first player removed 8 counters, thus allowing the second player to remove up to $2 \times 8 = 16$ counters. Incredibly, this doubling game $N_d\,(n, k)$ can be solved using *Fibonacci* representation. The number 100 has been represented as the sum of three Fibonacci numbers. If the first player removes 3 counters, $(100, 10) \mapsto (97, 6)$ the number of summands has been reduced to two $(97 = 89 + 8)$. Note that the second player can remove no more than 6, so that player cannot reduce the number of summands. In our paper [?], we prove that once a player in the game $N_d\,(n, k)$ has been able to reduce the number of summands (Fibonacci numbers), he will be able to do this repeatedly on all future turns. Thus the game might go as follows: $(100, 10) \mapsto (97, 6) = (89 + 8, 6) \mapsto (89 + 4, 8) \mapsto (89, 8) \mapsto (81, 16) = (55 + 21 + 5, 16) \mapsto (55 + 21, 10) \mapsto \ldots$. Note that each odd move beginning with the first one reduces the number of summands.

Just as one can express every position integer as a sum of distinct powers of 2, one can also write every such number as a sum of Fibonacci numbers. The method of repeated subtraction we discussed in section 1 works nicely. Pick the largest Fibonacci number not exceeding the given integer, then subtract the Fibonacci number and continue the process with the difference. For example, to write 100 in Fibonacci, list the Fibonacci numbers up to 100: $\{1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots\}$. Then subtract the largest member of the list from 100. So, we have $100 = 89 + 11$ and we continue the process with the obtained differences: $100 = 89 + 11 = 89 + 8 + 3$ which we could write as $1{\cdot}89+0{\cdot}55+0{\cdot}34+0{\cdot}21+0{\cdot}13+1{\cdot}8+0{\cdot}5+1{\cdot}3+0{\cdot}2+0{\cdot}1 = 1000010100_f$. The winning strategy for $N_d(n)$ is analogous to the winning strategy for $N_i(n)$, and is given by Theorem 2.

Theorem 2. *Consider the game $N_d(n)$ where $n = \sum_{j=1}^{j=l} e_j \cdot F_j$, where $F_1 = 1, F_2 = 2, \ldots,$ is the sequence of Fibonacci numbers. In other words, $e_{le_{l-1}} \cdots e_1$ is the Fibonacci representation of n. Then the next player has a winning move from the position (n, k) if and only if $F_t \leq k$ where $e_t = 1$ and for all $j < t$, $e_j = 0$. In other words, if the player can reduce the number of summands in the Fibonacci representation of n, then he can win. If not, he can't win.*

For proofs of these theorems, see the paper Dynamic One-Pile Nim, with Arthur Holshouser and James Rudzinski, in *The Fibonacci Quarterly*, a copy of which can be found at

`http://www.math.uncc.edu/~hbreiter/dynamic.pdf`