A stone-picking game

Nguyen Manh Tien- Tu Nguyen Thai Son

December 5, 2010

Team Information:

Affiliation:
High School for the Gifted
National University of Hochiminh city

Teacher/Coach/Mentor:
Tran Nam Dung, PhD, Hochiminh city University of Science

States/provinces, Country:
Hochiminh city, Vietnam

Abstract

Consider the following two-player game, in which players take turns removing stones from a heap of n one. On each turn,a player must remove k stones where k in a given "$rule - set$" S. If the amount of stones in the heap is lower than $minS$ then he can remove all.The winner is the one who takes the last stone.

Let f_S be a particular function of S and n, takes values from the set of {0,1}. $f_S(n) = 1$ if and only if the first one wins. Let $(a_0a_1a_2a_3\ldots)$ be a binary string with infinity length where $a_i = f(S, i)$. The string will be repeated (since the first or another term) with period $T(S)$. For convenient, let $B(S) = a_0a_1a_2a_3a_{T(S)-1}$ be that unique string of S. In this paper, we will have a research on $B(S)$ and $T(S)$.

For simple expression of the result and further research in various cases, we contruct a rational number $E(S)$ in $(0, 1)$ as the "$result - number$" of set S. It will be easier for further research if properties of $E(S)$ are considered.

Finally, we come up with the recurrent expression for the general case, in which various heaps of stones and their rule sets are considered. It is also a general problem of the Nim game.

Contents

I Project introduction and the recurrent expression 4

II Summary of our results 9

III Detailed Proofs 10

1 Case $S = \{a\}$ 10

2 Case $S = \{1, b\}$ 10
- 2.1 $S = \{1, b\}$ where b is odd 10
- 2.2 $S = \{1, b\}$ where b is even 10

3 Case $S = \{2, b\}$ 11
- 3.1 $S = \{2, b\}$ where $b = 4k$ or $4k + 1$ 11
- 3.2 $S = \{2, b\}$ where $b = 4k + 3$ 12
- 3.3 $S = \{2, b\}$ where $b = 4k + 2$ 12

4 Case $S = \{1, 2, c\}$ 13
- 4.1 $S = \{1, 2, 3\}$ 13
- 4.2 $S = \{1, 2, 4\}$ 14
- 4.3 $S = \{1, 2, 5\}$ 14
- 4.4 $S = \{1, 2, 6\}$ 15
- 4.5 $S = \{1, 2, c\}$ where $c = 3k \pm 1$ 16
- 4.6 $S = \{1, 2, c\}$ where $c = 3k$ 16

5 Case $S = \{1, 3, c\}$ 17
- 5.1 $S = \{1, 3, 4\}$ 17
- 5.2 $S = \{1, 3, 5\}$ 17
- 5.3 $S = \{1, 3, c\}$ where c is odd 17
- 5.4 $S = \{1, 3, c\}$ where c is even 18

6 Case $S = \{1, b, c\}$ where b is odd 19
- 6.1 $S = \{1, b, c\}$ where b and c are odd 19
- 6.2 $S = \{1, b, c\}$ where b is odd and c is even 19

IV Open problems 22

V References 23

Part I
Project introduction and the recurrent expression

We consider a two-player game, in which players take turns removing stones from a heap of n one. On each turn,a player must remove k stones where k in a given "$rule - set$" S. If the amount of stones in the heap is lower than $minS$ then he can remove all. The winner is the one who takes the last stone. Assume that both player are extremely wise and they have the best tactics for their turn. Let f_S be a particular function of S and n, takes values from the set of {0,1}. $f_S(n) = 1$ if and only if the first one wins.

In this paper, we pay more advantage on the case $S = \{a, b, c\}$, which is given as an example below:

Example: Consider a two-player game, in which players take turns removing stones from a heap of 2010 one. On each turn,a player must remove 3, 5, or 6 stones. If the amount of stones in the heap is lower than 3 then he can remove all. The winner is the one who takes the last stone. Who has the winning strategy and why?

Solution: We call a number n "$tactical$" if and only if when we replace 2010 by n, the first player has his winning tactics. Otherwise, if the second player has winning tactics, n is called $un - tactical$. It's trivial that $1, 2$ and 3 are $tactical$.

A number n is $tactical$ if and only if among $n-3$, $n-5$ and $n-6$ there is a $un - tactical$ number (because the first one can win the game by making the second one lose)

A number n is $un - tactical$ if and only if $n - 3$, $n - 5$ and $n - 6$ are ALL $tactical$ numbers (because however the first one picks, the heap remains a $tactical$ number of stones for the second one's turn)

Using induction, we will prove that k is $un - tactical$ if and only if k is divisible by 4. It's easy to check that 0 and 4 are $un-tactical$ while $1, 2, 3, 5, 6, 7$ are $tactical$.

Assume that the statement is true $\forall\ k \leq n$ $(n \geq 7)$.

If $n + 1$ is divisible by 4 then $n + 1 - 3$, $n + 1 - 5$, $n + 1 - 6$ is not divisible by 4 and all of them are $tactical$. So, in this case n+1 os $un - tactical$

If $n+1$ is not divisible by 4, among $n+1-3$, $n+1-5$ and $n+1-6$, there is exactly one number which is divisible by 4 -or $un - tactical$. That means $n + 1$ is $tactical$.

The result will follow. And because 2010 is not a multiple of 4, so the first player has his winning tactics.

In general, let $f(n)$ be a mapping from Z to $\{0,1\}$. $f(n)=0$ if and only if n is *tactical*, otherwise $f(n)=0$.

Moreover,

$$\begin{cases} f(n)=0\ if\ and\ only\ if\ f(n-3).f(n-5).f(n-6)=1 \\ f(n)=1\ if\ and\ only\ if\ f(n-3).f(n-5).f(n-6)=0 \end{cases}$$

which means

$$f(n)=1-f(n-3).f(n-5).f(n-6) \tag{1}$$

For more generality, if $S=\{a_1,...,a_k\}$ is the set of amount of stones that a player can pick on his turn then (1) becomes

$$f_S(n)=1-\prod_{i=1}^{k} f_S(n-a_i) \qquad \forall\ n \geq max\{S\} \tag{2}$$

And because if $n < a_i$ and $a_i \neq min\{S\}$ then $f_S(n)=f_{S\setminus\{a_i\}}(n)$ and if $0 < n < min\{S\}$ then $f(n)$=1, so, as a result, we can write

$$f_S(n)=1-\prod_{X\in S, X\leq n} f_S(n-X) \qquad \forall\ n \geq 0 \tag{3}$$

So we can assume that $f(n)=1 \quad \forall n < min\{S\}$ but $n \neq 0$, and (3) become

$$\begin{cases} f(0)=0; \\ f(n)=1 \qquad \forall n < min\{S\}, n \neq 0; \\ f(n)=1-\prod_{i=1}^{k} f(n-a_i) \qquad \forall\ n \geq min\{S\} \end{cases} \tag{4}$$

Form now then, we assume that $S=\{a_1,...,a_{|S|}\}$, where $a_i < ai+1$. We come to the definition of 'the result binary string'

Definition I.1: The binary string $B(S)=b_1b_2...$ with infinite length and $b_i=f_S(i)$ is called the *unique binary string* of S (or the *result binary string*)

Lemma I.1: There exists a number $0 \leq i \leq 2^{a_{|S|}}+a_{|S|}+1$ that $B(S)$ is repeated since the i^{th} bit

Proof: Since $C(n)=b_{n-1},b_{n-2},...,b_{n-a_{|S|}}$ is a finite-length binary string, so it can take just $2^{a_{|S|}}$ values while there are infinite $C(n)$. By Pigeonhole's principle, there exists positive integers $i < j \leq 2^{a_{|S|}}+a_{|S|}+1$ such that $C(i)=C(j)$.

Using induction and 4, we have $C(i+k)=C(j+k) \quad \forall k \geq 0$. And because $f_S(n)$ completely depend on $f_S(n-1), f_S(n-2), ..., f_S(n-a_{|S|})$, we can deduce that $C(n)=C(n+j-i) \quad \forall n \geq i$. In other words, $B(S)$ is repeated since the i^{th} bit.

Hence, in this paper $B(S)$ will be written in the form $b_0b_1...b_h < b_{h+1}...b_k >$ with $b_{h+1}...b_k$ is the repeated part.

Moreover, if we change the rule to "who picks the last stone is the loser", we got the same recurrent expression but with different beginning values. That is:

$$\begin{cases} f(0) = 1; \\ f(n) = 0 \qquad \forall n < min\{S\}, n \neq 0; \\ f(n) = 1 - \prod_{i=1}^{k} f(n - a_i) \qquad \forall\ n \geq min\{S\} \end{cases} \tag{5}$$

(4) is our key idea in building the algorithm to solve the general problem.

Our algorithm has:

Input : The set S, a number N

Output: The first $N + 1$ bits of $B(S)$

And the following is our source code (written in Pascal):

```
Program Stone_Picking;
Var
      j,e,d,i,f: integer;
      S: array[1..400] of byte;
      Ax: array[1..400] of byte;
      T: String;

Begin
       T:='Y';
       While T='Y' do
       Begin
            Write('Input the size of S : ');
            Readln(e);
            For i:=1 to e do
                Begin
                    Write('Input S[ ',i,' ] : ');
                    Readln(S[i]);
                End;
            Write('Input the length of binary string : ');
            Readln(d);
            Ax[200]:=0;
            For i:=1 to 199 do
                Begin
                    Ax[i]:=1;
                End;
            For i:=1 to d do
                Begin
                    If i<S[1] then Ax[i+200]:=1;
                    If i>=S[1] then
                        Begin
                            f:=1;
                            For j:=1 to e do
                                Begin
```

```
                                        f:=f*Ax[i+200-S[j] ]
                                  End;
                                  Ax[i+200]:=1-f
                              End;
                    End;
                For i:=0 to d do
                    Begin
                        Writeln( 'f( ', i:3 ,' ) = ',Ax[i+200]);
                    End;
                Write('Restart? <Y/N> : ');
                Readln(T);
            End;
End.
```

By the algorithm above, we can conclude that:

Conclusion I.1: If there exists a positive integer $T > max\{S\}$ that

$$f_S(i) = f_S(i+T) \quad \forall i \in [0, T]$$

then $f_S(n)$ is periodic with period T

Proof: We will use induction on i to prove the Conclusion

The Conclusion is true for all $i \in [0, T]$. Assume that it's also true for all $i \in [0, k]$ $(k \geq T)$, we have

$$f_S(k+1+T) = 1 - \prod_{x \in S} f_S(k+1+T-x) = 1 - \prod_{x \in S} f_S(k+1-x) = f_S(k+1)$$

(true because $k \geq T \geq x$)

So the Conclusion is also true for all $i \in [0, k+1]$. The result will follow then.

Now, we want to express the result in a simple way. Unlike using an binary string, which is very hard to remember, the following method uses only a rational number.

Definition I.2: For a set S with binary string $B(S) = x_1x_2 \ldots x_n \ldots$, we consider following real number as the "$result - number$" of S:

$$E(S) = \sum_{i=1}^{\infty} \frac{x_i}{2^i}$$

Lemma I.2: For all finite set S, its $result{-}number$ $E(S)$ is a rational number in $(0, 1)$

Proof: Because $B(S)$ can not be filled with only '0' or '1' from some term onward, then $0 = \sum_{i=1}^{\infty} \frac{0}{2^i} < E(S) < \sum_{i=1}^{\infty} \frac{1}{2^i} = 1$, which means $B(S)$ is in $(0, 1)$

By $Lemma I.1$, $B(S)$ can be expressed as the following form:
$B(S) = b_1 b_2 \dots b_k < a_1 a_2 \dots a_T >$. Then

$$E(S) = \sum_{i=1}^{k} \frac{b_i}{2^i} + \sum_{i=1}^{T} \frac{a_i}{2^{k+i}} (\sum_{j=0}^{\infty} (\frac{1}{2})^{Tj}) = \sum_{i=1}^{k} \frac{b_i}{2^i} + \frac{2^T}{2^T - 1} \sum_{i=1}^{T} \frac{a_i}{2^{k+i}}$$

which is rational.

Lemma I.3: For all result binary string $B(S)$, its $result - number$ $E(S)$ is unique, that is

$$E(S) = E(S') \Rightarrow B(S) = B(S')$$

Proof: Assume in contradiction that $B(S) = x_1 x_2 \dots \neq B(S') = y_1 y_2 \dots$. Let t be the smallest index that $x_t \neq y_t$. Without loss of generality, assume that $x_t = 1$ and $y_t = 0$. We have:

$$E(S) - E(S') = \frac{1}{2^t} + \sum_{i=t+1}^{\infty} \frac{x_i - y_i}{2^i} \geq \frac{1}{2^t} + \sum_{i=t+1}^{\infty} \frac{1}{2^i} = 0$$

The equaltion holds if and only if $x_i = 0$ adn $y_i = 1$ for all $i > t$, which is not true because $B(S)$ can not be filled with only '0' or '1' from some term onward. The contraction completes the proof.

Part II

Summary of our results

The following is our result in different cases

Lemmas and Conclusions in I

Case $S = \{a\}$

$B(S) =< 0\underbrace{11\dots1}_{abits}\underbrace{00\dots0}_{a-1bits}$ and $E(S) = \frac{2^a}{2^a+1}$

Case $S = \{1, b\}$

If b is odd, $B(S) =< 0, 1 >$ and $E(S) = \frac{2}{3}$

If b is even, $B(S) =< \underbrace{0101..01}_{b\ bits}1 >$ and $E(S) = \frac{4(2^b+1)}{3(2^{b+1}-1)}$

Case $S = \{2, b\}$

If $b \equiv 0\ (mod\ 4)$, $B(S) =< \underbrace{01100110..0110}_{b\ bits}01 >$ and $E(S) = \frac{2(2^{b+3}-3)}{5(2^{b+2}-1)}$

If $b \equiv 1\ (mod\ 4)$, $B(S) =< \underbrace{01100110..0110}_{b-1\ bits}011 >$ and $E(S) = \frac{2(2^{b+3}-1)}{5(2^{b+2}-1)}$

If $b \equiv 2\ (mod\ 4)$, $B(S) =< 0110 >$ and $E(S) = \frac{4}{5}$

If $b \equiv 3\ (mod\ 4)$, $B(S) =< \underbrace{01100110..0110}_{b-3\ bits}01110 >$ and $E(S) = \frac{2(2^{b+2}+3)}{5(2^{b+2}-1)}$

Case $S = \{1, 2, c\}$

If $b \equiv \pm1\ (mod\ 3)$, $B(S) =< 011 >$ and $E(S) = \frac{6}{7}$

If $b \equiv 0\ (mod\ 3)$, $B(S) =< \underbrace{011011...011}_{b\ bits}1 >$ $E(S) = \frac{2(3.2^{c+1}+1)}{7(2^{c+1}-1)}$

Case $S = \{1, b, c\}$ **where** b **is odd**

If c is odd, $B(S) =< 0, 1 >$ and $E(S) = \frac{2}{3}$

If c is even, $B(S) =< \underbrace{0101..01}_{c\ bits}\underbrace{11..1}_{b\ bits} >$ and $E(S) = \frac{2^{b+c}+2^{b+1}-3}{3(2^{b+c}-1)}$

The recurrent expression (8)

Part III
Detailed Proofs

1 Case $S = \{a\}$

As we have proved before:

$$\begin{cases} f_S(n) = 1 \quad\quad for \quad n = 1, 2, ..., a-1 \\ f_S(n) = 1 - f_S(n-a) \quad\quad \forall n \geq a \end{cases}$$

Since $f_S(n) = 1 - f_S(n-a) = 1 - (1 - f(n-2a)) = f(n-2a)$, f_S is repeated with the period of $2a$

In brief, we got the following result:

Conclusion 1.1: For all positive integer a, $f_S(n)$ where $S = \{a\}$ is defined as following:

$$\begin{cases} f_S(n) = 1 \quad\quad for\ n \equiv 1, 2, ..., a \quad\quad (mod\ 2a) \\ f_S(n) = 0 \quad\quad for\ n \equiv a+1, a+2, ..., 2a \quad\quad (mod\ 2a) \end{cases}$$

And $E(S) = \frac{2^a}{2^a+1}$

2 Case $S = \{1, b\}$

2.1 $S = \{1, b\}$ where b is odd

We can easily check that $f_{(1,b)}(0) = 0$ and $f_{(1,b)}(1) = 1$
Moreover, $f_{(1,b)}(n) = 1 - f_{(1,b)}(n-1).f_{(1,b)}(n-a)$,

Note that if n is odd, because players can take just an odd amount of stones, after the second man's turn, the amount of stones remains odd, so the first man always win the game.

If n is even, after the first man's turn, the amount of stones remains odd, so the second man is the winner of game.

So, the result binary string in this case is $< 0, 1 >$ and $E(S) = \frac{2}{3}$.

2.2 $S = \{1, b\}$ where b is even

For simple writing, let $f(n)$ be $f_{(1,b)}(n)$.

We can easily check that $f(0) = 0$ and $f(1) = 1$.
Moreover, $f(n) = 1 - f(n-1).f(n-b)$

If $n < b$ two man have only the way to pick stones ,so if n is odd, the first player wins and if n is even , the second one wins.
So when $n = 0, 1, 2..., b-1$
$f(n) = 0, 1, 0, 1, 0, 1, ...0, 1$ and when $n = b$, $f(n) = 1$.

We prove that the binary String of this case has period of $a+1$
If $n = b+1, b+2, ..., 2b+1$ we prove that $f(n) = 0,1,0,1...,0,1,1$
We can easily check that $f(b+1) = 0$ and $f(b+2) = 1$

Using induction, we'll prove that $f((b+1)+k) = f(k)$ for all $k = 0, 1, ..b$
Assume that $f((b+1)+k) = f(k)$ is true for k, we prove that $f(b+1)+k+1) = f(k+1)$.

We have $f(n) = 1 - f(n-1).f(n-b)$
$\Rightarrow f((b+1)+k+1) = 1 - f((b+1+k).f(k+2) = 1 - f(k).f(k+2)$
$\Rightarrow f(b+1+k+1) = 1 - f(k).f(k+2)$
If $k+2 \leq b-1$ we have $f(k) = f(k+2)$ so $f(b+1+k+1) = 1 - f(k)$
Note that $f(k+1) = 1 - f(k)$ so we have $f(b+1+k+1) = f(k+1)$

So when $N = b+1, b+2, ...2b$ $f(N) = 0,1,0,1,...0,1$

Finally, we have $f(2b+1) = 1 - f(2b).f(b+1) = 1$

In brief,in this case we have binary String is $0,1,0,1,...,0,1,1$ and $E(S) = \frac{4(2^b+1)}{3(2^{b+1}-1)}$.

3 Case $S = \{2, b\}$

In this case, the function $f_{(2,b)}$ satisfies the expression:

$$f_{(2,b)}(n) = 1 - f_{(2,b)}(n-2).f_{(2,b)}(n-b) \quad \forall n \geq b$$

The result can be divided into 3 sub-cases:

3.1 $S = \{2, b\}$ where $b = 4k$ or $4k+1$

We have the following conclusion:

Conclusion: 3.1.1 If b has the form of $4k$ or $4k+1$ then $f_{(2,b)}(n) = f(n)$, where $f(n)$ is defined as below:

$$\begin{cases} f(n) = 0 \quad if \quad n \equiv 3 \quad or \quad n \equiv 0(mod4); \quad n \leq b \\ f(n) = 1 \quad if \quad n \equiv 1 \quad or \quad n \equiv 2(mod4); \quad n \leq b \\ f(b) = 1; \qquad f(b+1) = 1 \\ f(n) = f(n+b+2) \qquad \forall n \geq 0 \end{cases}$$

Proof: It's obvious that the function $f_{(2,b)}$ is unique, so we just have to prove that $f(n)$ satisfies all conditions for $f_{(2,b)}$ and complete the proof.

Indeed, we have:

1. $f_{(2,b)}(n) = f_{(2)}(n) = f(n) \quad \forall n \leq b-1$ and $f_{(2,b)}(b) = f(b) = 1$
2. $b-1 \equiv 0$ or $3(mod4)$, so $f(b+1-2) = 0$, leads to $f(b+1) = 1$

3. We'll prove that $f(n) = 1 - f(n-2).f(n-b)$. We can check that $f(n)$ above has the following properties:

$$\begin{cases} f(n) = 0 \to f(n \pm 2) = 0 \\ f(n+2) = f(n-2) = 1 \to f(n) = 0 \end{cases} \tag{6}$$

then we have

$$f(n) = 1 - f(n-2).f(n+2) = 1 - f(n-2)f(n-b)$$

And the conclusion will follow.

So, in this case, the result binary string has the minimum period defined by: the first $b-1$ bits as in the case $S = \{2\}$, and then added a bit '1'. And the $result - number$ is $\frac{2(2^{b+3}-3)}{5(2^{b+2}-1)}$

3.2 $S = \{2, b\}$ where $b = 4k + 3$

We have the following conclusion:

Conclusion: 3.2.1 If b has the form of $4k+3$ then $f_{(2,b)}(n) = f(n)$, where $f(n)$ is defined as below:

$$\begin{cases} f(n) = 0 \quad if \quad n \equiv 3 \quad or \quad n \equiv 0 (mod 4); \quad n \leq b \\ f(n) = 1 \quad if \quad n \equiv 1 \quad or \quad n \equiv 2 (mod 4); \quad n \leq b \\ f(b) = 1; \qquad f(b+1) = 0 \\ f(n) = f(n+b+2) \qquad \forall n \geq 0 \end{cases}$$

Proof: It's obvious that the function $f_{(2,b)}$ is unique, so we just have to prove that $f(n)$ satisfies all conditions for $f_{(2,b)}$ and complete the proof.

Indeed, we have:

1. $f_{(2,b)}(n) = f_{(2)}(n) = f(n) \quad \forall n \leq b-1$ and $f_{(2,b)}(b) = f(b) = 1$

2. $b - 1 \equiv 2(mod4)$, so $f(b+1-2) = 1$ and $f(b+1-b) = f(1) = 1$, leads to $f(b+1) = 0$

3. We can prove, as the same way in 3.1, that f satisfies (6)

And the conclusion will follow.

So, in this case, the result binary string has the minimum period defined by: the same first $b-1$ bits as in the case $S = \{2\}$, and then added a bit '0'. Hence, $E(S) = \frac{4(2^{b+2}+3)}{5(2^{b+2}-1)}$

3.3 $S = \{2, b\}$ where $b = 4k + 2$

We have the following conclusion:

Conclusion: 3.3.1 If b has the form of $4k+2$ then $f_{(2,b)}(n) = f_{(2)}(n)$

Proof: It's obvious that the function $f_{(2,b)}$ is unique, so we just have to prove that $f_{(2)}(n)$ satisfies all conditions for $f_{(2,b)}$ and complete the proof.

Notice that $f_{(2,b)}(n) = f_{(2)}(n) \quad \forall n \leq b-1$ and $f_{(2,b)}(b) = 1f_{(2)}(b)$, we only need to prove that

$$f_{(2)}(n) = 1 - f_{(2)}(n-2).f_{(2)}(n-b) \quad \forall n \geq b$$

The expression above is trivial because $f_{(2)}$ has the period of 4, so:

$$f_{(2)}(n) = 1 - f_{(2)}(n-2) = 1 - f_{(2)}^2(n-2) = 1 - f_{(2)}(n-2).f_{(2)}(n-b)$$

So, in this case, the result binary string has the minimum period defined of $< 0110 >$ and $E(S) = \frac{4}{5}$, as same as in the case $S = 2$.

4 Case $S = \{1, 2, c\}$

For simple writing, let $f(n)$ be $f_{(1,2,c)}(n)$.

4.1 $S = \{1, 2, 3\}$

Lemma 4.1.1: The function f define above satisfies the following expression:

$$f(4t + k) = f(k) \qquad \forall t \in N \textit{ and } k = 0, 1, 2, 3$$

Proof: Indeed, when $n = 0, 1, 2, 3, 4, 5, 6, 7$ we have $f(n) = 0, 1, 1, 1, 0, 1, 1, 1$

We'll use induction on t for the proof. It's trivial if $t = 1$

Assume that $f(4t + k) = f(k)$ for any $t \in N$ and $k = 0, 1, 2, 3$

We have :

$f(4(t+1)+k) = 1 - f(4(t+1)+k-1).f(4(t+1)+k-2).f(4(t+1)+k-3)$
$= 1 - f(4t+k+3).f(4t+k+2).f(4t+k+1)$
$= 1 - f(k+3).f(k+2).f(k+1)$

If $k = 0$ we have:$f(4(t+1)) = 1 - f(3).f(2).f(1) = 0 = f(0)$

If $k = 1$ we have:$f(4(t+1)+1) = 1 - f(4).f(3).f(2) = 1 = f(1)$

If $k = 2$ we have:$f(4(t+1)+2) = 1 - f(5).f(4).f(3) = 1 = f(2)$

If $k = 3$ we have:$f(4(t+1)+3) = 1 - f(6).f(5).f(4) = 1 = f(3)$

And the statement is also true for $t + 1$.

In brief,we have $f(4t + k) = f(k)$ for all $t, k \in N$

So we have the binary string is $< 0, 1, 1, 1 >$ and the period is 4.

4.2 $S = \{1, 2, 4\}$

Lemma 4.2.1: The function f define above satisfies the following expression:

$$f(3t + k) = f(k) \qquad \forall t \in N \ and \ k = 0, 1, 2$$

Proof: When $n = 0, 1, 2, 3, 4, 5$ we have $f(n) = 0, 1, 1, 0, 1, 1$

We'll use induction on t for the proof. It's trivial if $t = 1$

Assume that $f(3t + k) = f(k)$ for any $t \in N$ and $k = 0, 1, 2$

We have :

$f(3(t + 1) + k) = 1 - f(3(t + 1) + k - 1).f(3(t + 1) + k - 2).f(3(t + 1) + k - 3)$
$= 1 - f(3t + k + 2).f(3t + k + 1).f(3t + k)$
$= 1 - f(k + 2).f(k + 1).f(k)$

If $k = 0$ we have:$f(3(t + 1)) = 1 - f(2).f(1).f(0) = 0 = f(0)$

If $k = 1$ we have:$f(3(t + 1) + 1) = 1 - f(3).f(2).f(1) = 1 = f(1)$

If $k = 2$ we have:$f(3(t + 1) + 2) = 1 - f(4).f(3).f(2) = 1 = f(2)$

And the statement is also true for $t + 1$.

In brief,we have $f(3t + k) = f(k)$ for all $t, k \in N$

So we have the binary string is $< 0, 1, 1 >$ and the period is 3.

4.3 $S = \{1, 2, 5\}$

Lemma 4.3.1: The function f define above satisfies the following expression:

$$f(3t + k) = f(k) \qquad \forall t \in N \ and \ k = 0, 1, 2$$

Proof: When $n = 0, 1, 2, 3, 4, 5$ we have $f(n) = 0, 1, 1, 0, 1, 1$

We'll use induction on t for the proof. It's trivial if $t = 1$

Assume that $f(3t + k) = f(k)$ for any $t \in N$ and $k = 0, 1, 2$

We have :

$f(3(t + 1) + k) = 1 - f(3(t + 1) + k - 1).f(3(t + 1) + k - 2).f(3(t + 1) + k - 3)$
$= 1 - f(3t + k + 2).f(3t + k + 1).f(3t + k)$
$= 1 - f(k + 2).f(k + 1).f(k)$

If $k = 0$ we have:$f(3(t + 1)) = 1 - f(2).f(1).f(0) = 0 = f(0)$

If $k = 1$ we have:$f(3(t + 1) + 1) = 1 - f(3).f(2).f(1) = 1 = f(1)$

If $k = 2$ we have:$f(3(t + 1) + 2) = 1 - f(4).f(3).f(2) = 1 = f(2)$

And the statement is also true for $t+1$.

In brief,we have $f(3t+k) = f(k)$ for all $t, k \in N$

So we have the binary string is $< 0, 1, 1 >$ and the period is 3.

4.4 $S = \{1, 2, 6\}$

Lemma 4.4.1: The function f define above satisfies the following expression:

$$f(7t+k) = f(k) \qquad \forall t \in N \textit{ and } k = 0, 1, 2, 3, 4, 5, 6$$

Proof: When $n = 0, 1, 2., .., 12, 13$ we have $f(n) = 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1$
We'll use induction on t for the proof. It's trivial if $t = 1$
Assume that $f(7t+k) = f(k)$ for all $t \in N$ and $k = 0, 1, 2, 3, 4, 5, 6$
We have :
$f(7(t+1)+k) = 1 - f(7(t+1)+k-1).f(7(t+1)+k-2).f(7(t+1)+k-3)$
$= 1 - f(7t+k+6).f(7t+k+5).f(7t+k+4)$
$= 1 - f(k+6).f(k+5).f(k+4)$

If $k = 0$ we have:$f(7(t+1)) = 1 - f(6).f(5).f(4) = 0 = f(0)$

If $k = 1$ we have:$f(7(t+1)+1) = 1 - f(7).f(6).f(5) = 1 = f(1)$

If $k = 2$ we have:$f(7(t+1)+2) = 1 - f(8).f(7).f(6) = 1 = f(2)$

If $k = 3$ we have:$f(7(t+1)+3) = 1 - f(9).f(8).f(7) = 0 = f(3)$

If $k = 4$ we have:$f(7(t+1)+4) = 1 - f(10).f(9).f(8) = 1 = f(4)$

If $k = 5$ we have:$f(7(t+1)+5) = 1 - f(11).f(10).f(9) = 1 = f(5)$

If $k = 6$ we have:$f(7(t+1)+6) = 1 - f(12).f(11).f(10) = 1 = f(6)$

In brief,we have $f(7t+k) = f(k)$ for all $t, k \in N$

And the statement is also true for $t+1$.

So we have the binary string is $< 0, 1, 1, 0, 1, 1, 1, >$ and the period is 7.
It make us predict if $c = 3k$ then the period is $c+1$ and the binary string is $< 0, 1, 1, 0, 1, 1, , 0, 1, 1, 1 >$ and when $c = 3k \pm 1$ the binary string is $< 0, 1, 1 >$ and $E(S) = \frac{6}{7}$.

4.5 $S = \{1, 2, c\}$ where $c = 3k \pm 1$

We have $f(n) = 1 - f(n-1).f(n-2).f(n-c)$
We can easily check that $f(0) = 0$ and $f(1) = 1,\ f(2) = 1$,$f(3) = 0$.
We use induction to prove the proposition :$f(3k \pm 1) = 1$,$f(3k) = 0$
From $f(n) = 1 - f(n-1).f(n-2).f(n-c)$
If $n = 3k$ then $f(n-1), f(n-2), f(n-c) = 1$ so $f(n) = 0$
If $N = 3k+1$ then $f(n-1) = 0 \Longrightarrow f(n) = 1$
If $n = 3k+2$ then $f(n-2) = 0 \Longrightarrow f(n) = 1$
In brief, the proposition is true.

4.6 $S = \{1, 2, c\}$ where $c = 3k$

Assume that ≥ 3.
We can easily check that $f(0) = 0$ and $f(1) = 1,\ f(2) = 1$,$f(3) = 0$.
For all $k \leq c-1$, two player only have pick 1 or 2 , we prove that for all $k \leq c-1$, $f(k) = 0$ if $k = 3t$ and $f(k) = 1$ if $k = 3t \pm 1$

It is easily because the case $(1, 2)$ we have for all $k \leq c-1$, $f(k) = 0$ if $k = 3t$ and $f(k) = 1$ if $k = 3t \pm 1$

When $k = c$ we have $f(c) = 1$
So, when $N = 0, 1, 2, ...c$ we have $f(N) = 0, 1, 1, 0, 1, 1, ..., 0, 1, 1, 1$

We prove that for all $N = c+1, c+2, ...2c+1$ we have $f(N) = 0, 1, 1, 0, 1, 1, ..., 0, 1, 1, 1$
This problem is equivalent with for all $k = 0, 1, 2, ...c$ we have $f(c+1+k) = f(k)$
We prove the proposition for all $k = 0, 1, 2, ...c$ we have $f(c+1+k) = f(k)$
We can easily check that :
$f(c+1) = 1 - f(c).f(c-1).f(1) = 0$
$f(c+2) = 1 - f(c+1).f(c).f(2) = 1$

Assume that the proposition is true for all $1 \leq k \leq c-1$
We have $f(c+1+k+1) = 1 - f(c+1+k).f(c+1+k-1).f(k)$
Follow the assumption, we have:$f(c+1+k) = f(k)$ and $f(c+1+k-1) = f(k-1)$
$\Longrightarrow f(c+1+k+1) = 1 - f(k).f(k-1)$
If $k = 3t$ we have $f(c+1+k+1) = 1 = f(k+1)$
If $k = 3t+1$ we have $f(c+1+k+1) = 1 = f(k+1)$
If $k = 3t+2$ we have $f(c+1+k+1) = 0 = f(k+1)$
So we have the proposition is true for all $1 \leq k \leq c-1$
Finally, we have $f(c+1+c) = 1 - f(2c).f(2c-1).f(c+1) = 1$
$\Longrightarrow$ the proposition is true for all $k = 0, 1, 2..., 2c+1$

In brief, the binary string in this case is $< 0, 1, 1, 0, 1, 1, ..., 0, 1, 1, 1 >$ and $E(S) = \frac{2(3.2^{c+1}+1)}{7(2^{c+1}-1)}$.

5 Case $S = \{1, 3, c\}$

For simple writing, let $f(n)$ be $f_{(1,3,c)}(n)$.

5.1 $S = \{1, 3, 4\}$

When $n = 0, 1, 2...$ we have $f(n) = 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1...$

It make us predict the binary string in this case is $< 0, 1, 0, 1, 1, 1, 1 >$ and the period is 7.

Note that we have $f(n) = 1 - f(n-1).f(n-3).f(n-4)$,
Simple induction complete the proof.

5.2 $S = \{1, 3, 5\}$

When $n = 0, 1, 2...$ we have $f(n) = 0, 1, 0, 1, ...$
It make us predict the binary string in this case is $< 0, 1 >$ and the period is 2.

Note that if n is odd, because players can take just an odd amount of stones, after the second player's turn, the amount of stones remains odd, so the first one always win the game.
If n is even, after the first man's turn, the amount of stones remains odd, so the second man is the winner of game. In brief, we have the binary string in this case is $< 0, 1 >$ and the period is 2.

5.3 $S = \{1, 3, c\}$ where c is odd

We prove that the binary string in this case is $< 0, 1 >$ and the period is 2.
Specific, we prove $f(2k) = 0$ and $f(2k+1) = 1$

We can easily check that $f(0) = 0$ and $f(1) = 1$.

Let x, y, z is the number of times that 1,3,c stones are picked we have:

$N = x.1 + y.3 + z.c = (x + y + z) + 2.y + (c - 1).z$

Note that $n - 1$ is even, we inferred :
If n is even, we have $x + y + z$ is even $\Rightarrow$ After a even number of turns, the stone will equal to 0 $\Rightarrow$ the second man is the winner because one who picks the last stone is the second man $\Rightarrow f(2k) = 0$
If n is odd, we have $x + y + z$ is odd $\Rightarrow$ After a odd number of turns, the stone will equal to 0 $\Rightarrow$ the first man is the winner because one who picks the last stone is the first man $\Rightarrow f(2k+1) = 0$

5.4 $S = \{1, 3, c\}$ where c is even

We can easily check that $f(0) = 0$ and $f(1) = 1$.
We have : $f(n) = 1 - f(n-1).f(n-3).f(n-c)$

For all $0 \le n \le c-1$, two player only have pick 1 or 3, so after the first man and the second man pick , the amount of stones' parity remains unchanged, we infer that if n is odd, the first man is the winner, and if n is even, the second man is the winner.

In brief, when $n = 0, 1, 2, ..., c-1$ we have $f(n) = 0, 1, 0, 1, ...1$ because $c-1$ is odd.

Next, we have $f(c) = 1$, and:
$f(c+1) = 1 - f(c).f(c-2).f(1) = 1$ because $c-2$ is even
$f(c+2) = 1 - f(c+1).f(c-1).f(2) = 1$

So when $n = 0, 1, 2, ...c+2$ we have $f(n) = 0, 1, 0, 1, ...0, 1, 1, 1, 1$

We use induction to prove that for $n = c+3, c+4, ...2c+2$, we have:
$f(n) = f(n-(c+3))$, in other words $f((c+3)+k) = f(k)$ for all $k = 0, 1, ...c+2$

$f(c+3) = 1 - f(c+2).f(c).f(3) = 0$

$f(c+4) = 1 - f(c+3).f(c+1).f(4) = 1$

$f(c+5) = 1 - f(c+4).f(c+2).f(5) = 1$

Assume that $f(c+3+k) = f(k)$ is true for any $k \ge 2$ we prove that $f(c+3+k+1) = f(k+1)$.

We have $f(c+3+k+1) = 1 - f(c+3+k).f(c+3+k-2).f(k+4)$. By the assumption, we have $f(c+3+k-2) = f(k-2)$ so
$f(c+3+k+1) = 1 - f(c+3+k).f(k-2).f(k+4)$

Note that $k-2$ and $k+4$ have the same parity, so if $k \le c-7$ we have $k+4 \le c-3$ so $f(k-2) = f(k+4)$.

If k is even, we have
$f(k) = f(k-2) = f(k+4) = 0$ so $f(c+3+k+1) = 1 = f(k+1)$

If k is odd, we have
$f(k) = f(k-2) = f(k+4) = 1$ so $f(c+3+k+1) = 1 - f(c+3+k) = 1 - f(k) = 0$

In brief, we have $f(c+3+k) = f(k) \quad \forall k = 0, 1, 2...c-7$
We can easily check that:

$f(c+3+c-6) = 1 - f(c+3+c-7).f(c+3+c-9).f(c-3)$
$= 1 - f(c-7).f(c-9).f(c-3) = 0 = f(c-6)$

$f(c+3+c-5) = 1 - f(c+3+c-6).f(c+3+c-8).f(c-2) = 1 = f(c-5)$

$f(c+3+c-4) = 1 - f(c+3+c-5).f(c+3+c-7).f(c-1)$
$= 1 - f(c-5).f(c-7).f(c-1) = 0 = f(c-4)$

$f(c+3+c-3) = 1 - f(c+3+c-4).f(c+3+c-6).f(c) = 1 = f(c-3)$

$f(c+3+c-2) = 1 - f(c+3+c-3).f(c+3+c-5).f(c+1) = 0 = f(c-2)$

$f(c+3+c-1) = f(c+3+c) = f(c+3+c+1) = f(c+3+c+2) = 1$

In brief, we have for all $n = c+4, ...2c+2, f(n) = 0, 1, 0, 1...., 0, 1, 1, 1, 1$
By induction, we have $f(t(c+3)+k) = f(k)$ for all $k = 0, 1, ...c+2$ and for all t$\in N$
So the binary string in this case is $< 0, 1, 0, 1..., 0, 1, 1, 1, 1 >$ and the period is $c+3$.

6 Case $S = \{1, b, c\}$ where b is odd

Because, as we have considered before, in the case $S = 1, b$ it's easier if b is odd, we just pay more advantage on the case that b is odd

6.1 $S = \{1, b, c\}$ where b and c are odd

It's trivial that the function $f_{(1,b,c)}(n)$ defined by:

$$\begin{cases} f_{(1,b,c)}(0) = 0 \\ f_{(1,b,c)}(1) = f_{(1,b,c)}(-n) = 1 \qquad \forall\ n \in Z^+ \\ f_{(1,b,c)}(n) = 1 - f_{(1,b,c)}(n-1).f_{(1,b,c)}(n-b).f_{(1,b,c)}(n-c) \quad \forall\ n \geq 1 \end{cases}$$

is unique.
So, we'll prove that $f(n) = \frac{1-(-1)^n}{2}$, which means $f(n) = 1$ if n is odd and $f(n) = 0$ if n is even, satisfies all conditions above. We can easily check that $f(0) = 0$ and $f(1) = 1$.

Moreover, $f(n) = 1 - f(n-1).f(n-b).f(n-c) \quad \forall n \geq 1$ is true because n and $n-1, n-b, n-c$ has different parity so $f(n)$ and $f(n-1), f(n-b), f(n-c)$ take different values from the set $\{0, 1\}$ So we can conclude that in the case $S = \{1, b, c\}$ where b and c is odd:

$$f_{(1,b,c)}(n) = f(n) = \frac{1-(-1)^n}{,} E(S) = \frac{2}{3}2$$

6.2 $S = \{1, b, c\}$ where b is odd and c is even

The problem becomes more and more complex because of the complexity of its result.

For example, in the case $S = \{1, 3, 7\}$ the output string returns $< 01010101111 >$, which has last 4 bits are '1111'
The following is some trivial fact that we can observe while approaching the problem:

1. For all $n \le c - 1$, $f_{(1,b,c)}(n) = f_{(1,b)}(n)$ (because $f_{(1,b,c)}(n - c) = 1$ so

$$f_{(1,b,c)}(n) = 1 - f_{(1,b,c)}(n-1).f_{(1,b,c)}(n-b).f_{(1,b,c)}(n-c) = f_{(1,b)}(n)).$$

So, in $[0, c-1]$, $f_{(1,b,c)}$ altenatively takes value of '0' and '1'.

2. Some special value of $f_{(1,b,c)}$: $f_{(1,b,c)}(c-1) = f_{(1,b,c)}(c) = 1$ because c is even

Now, we'll prove that "For all $b \le n \le b + c - 1$, $f_{(1,b,c)}(n) = 1$".

Lemma 1: For all $c \le n \le b + c - 1$,

$$f_{(1,b,c)}(n) = 1$$

where b is odd and c is even.

Proof: For simple writing, let $F(n)$ be $f_{(1,b,c)}(n)$.
Because $c \le n \le b + c - 1$, we can infer that $n - b$ and $n - c$ belong to $[0, c - 1]$. And because they have different parity, $F(n - b) \ne F(n - c)$, which deduces $F(n-b).F(n-c) = 0$ (F can take just 2 kinds of values).

The result will follow because of the expression

$$F(n) = 1 - F(n-1).F(n-b).F(n-c) \tag{7}$$

It could be infer from the Lemma 1 that $F(b+c) = 1 - F(b+c-1).F(b).F(c) = 0$
Our next step is proving that:
"For all $b + c \le n \le b + 2c - 1$, $f_{(1,b,c)}(n) = f_{(1,b,c)}(n - b - c)$"

Lemma 2: For all $b + c \le n \le b + 2c - 1$,

$$f_{(1,b,c)}(n) = f_{(1,b,c)}(n - b - c)$$

where b is odd and c is even.

Proof: For simple writing, let $F(n)$ be $f_{(1,b,c)}(n)$. The aim of the Lemma 2 is to prove that the output binary string is alternative with '0' and '1' from the $b + c^{th}$ bit to the $b + 2c - 1^{th}$ bit
It's true that F(b+c)=F(0). We will prove:

$$F(n) = 1 - F(n-1) \quad \forall\, n \in [b + c + 1, b + 2c - 1]$$

and everythings will follow.

It's obvious that $n-1 > n-b > n-c$ and $n-b \in [c, b+c-1] \cup [b+c, b+2c-1]$.

If $n-b \in [c, b+c-1]$ then $F(n-b)$ and $F(n-1).F(n-b) = F(n-1)$.

Otherwise $n-b \in [b+c, b+2c-1]$, because $n-1$ and $n-b$ are in $[b+c, b+2c-1]$ and have the same parity, by simple induction, we infer that $F(n-1) = F(n-b)$ and $F(n-1).F(n-b) = F(n-1)^2 = F(n-1)$

And in all cases, $F(n-1).F(n-b) = F(n-1)$

Moreover, because $n \in [b+c, b+2c-1]$, $n-c \in [0, c-1] \cup [c, b+c-1]$.

If $n-c \in [c, b+c-1]$ then $F(n-1).F(n-b).F(n-c) = F(n-1)$ (because $F(n-c) = 1$).

Otherwise $n-c \in [0, c-1]$ while $n-b \in [b+c-1, b+2c-1]$. Because $|[c, b+c-1] \cap Z|$ is odd, so if $n-c$ and $n-b$ have different parity then $F(n-c) = F(n-b)$. Hence, $F(n-c) = F(n-b)$ and $F(n-1).F(n-b).F(n-c) = F(n-1).F(n-b)^2 = F(n-1).F(n-b) = F(n-1)$. And in all cases, $F(n-1).F(n-b).F(n-c) = F(n-1)$.

The result is then follow.

Now we come to the most important result:

Lemma 3: For all $n \geq b+c$,

$$f_{(1,b,c)}(n) = f_{(1,b,c)}(n-b-c)$$

where b is odd and c is even.

Proof: The statement is true, as we have proved above, for all $b+c \leq n \leq b+2c-1$

If the statement is true for $n \leq k$ with $k \geq b+2c-1$. We'll prove that it's also true for $n = k+1$.

Indeed,

$$f_{(1,b,c)}(k+1) = 1 - f_{(1,b,c)}(k).f_{(1,b,c)}(k+1-b).f_{(1,b,c)}(k+1-c)$$

$$= 1 - f_{(1,b,c)}(k-b-c).f_{(1,b,c)}(k+1-2b-c).f_{(1,b,c)}(k+1-b-2c) = f_{(1,b,c)}(k-b-c)$$

Conclusion: In this case, the binary string is $< 010101...011...1 >$ with the length of $b+c$ and the last $b+1$ bits are '11...1'. And $E(S) = \frac{2^{b+c}+2^{b+1}-3}{3(2^{b+c}-1)}$

Part IV
Open problems

For more generality, we have the recurrent expression in the case that there are N heaps with the amount of stones are $a_1, a_2, ..., a_N$ and in the i^{th} heap, player can pick an number of stones in S_i.

By the same way we have done before, the recurrent expression of $f_{(S_1,...,S-N)}(a_1, ..., a_N)$ (for simple writing, we call it F) is

$$F(a_1, ..., a_N) = 1 - \prod_{i=1}^{N} \prod_{x \in S_i} F(a_1, ..., a_i - x, ..., a_N) \tag{8}$$

with the suitable beginning values (that by the reason of lacking time, we couldn't find out) that could make the algorithm works

In the case $S_1 = S_2 = ... = S_N = T$, the above problem becomes the Nim game, which is said to have originated in China, named and developed by Charles L. Bouton.[1]

However, solving the recurrent expression (8) is really complex, so we left it as an open problem for our paper.

Part V
References

1. The Nim game- Wikipedia.org, http://en.wikipedia.org/wiki/Nim