The Cookie Monster Problem

Leigh Marie Braswell

Under Mentorship of Tanya Khovanova

PRIMES Conference 2013

The Problem

- Cookie Monster is always hungry.

The Problem

- Cookie Monster is always hungry.
- We present him with a set of cookie jars, each filled with some number of cookies.

The Problem

- Cookie Monster is always hungry.
- We present him with a set of cookie jars, each filled with some number of cookies.
- He wants to empty them as quickly as possible. But...

The Problem

- Cookie Monster is always hungry.
- We present him with a set of cookie jars, each filled with some number of cookies.
- He wants to empty them as quickly as possible. But...
- On each of his moves, he must choose a subset of the jars and take the same number of cookies from each.

Example

16	10	6	3

Example

16 10 6 3

-10

-10

Example

16	10	6	3
-10 	-10 		
6	0	6	3

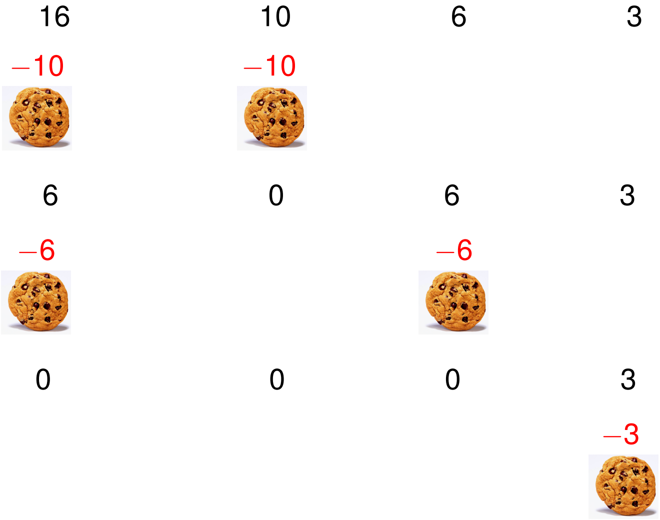
Example

16	10	6	3
-10 	-10 		
6	0	6	3
-6 		-6 	

Example

16	10	6	3
-10	-10		
			
6	0	6	3
-6		-6	
			
0	0	0	3

Example



Example

16	10	6	3
-10	-10		
			
6	0	6	3
-6		-6	
			
0	0	0	3
			-3
			
0	0	0	0

Initial Observations

- Cookie Monster may perform his moves in any order.

Initial Observations

- Cookie Monster may perform his moves in any order.
- Jars with equal number of cookies may be treated the same.

Initial Observations

- Cookie Monster may perform his moves in any order.
- Jars with equal number of cookies may be treated the same.
- Jars may be "emptied" without being emptied.

Initial Observations

- Cookie Monster may perform his moves in any order.
- Jars with equal number of cookies may be treated the same.
- Jars may be "emptied" without being emptied.
- Is there a procedure the monster can follow that will always lead to the optimal solution?

General Algorithms

- *Empty the Most Jars Algorithm*: the monster reduces the number of distinct jars by as many as he can for each move.

General Algorithms

- *Empty the Most Jars Algorithm*: the monster reduces the number of distinct jars by as many as he can for each move.
- $\{7, 4, 3, 1\}$

General Algorithms

- *Empty the Most Jars Algorithm*: the monster reduces the number of distinct jars by as many as he can for each move.
- $\{7, 4, 3, 1\}$
- $\xrightarrow[\text{Jar 1, Jar 2}]{-4} \{3, 0, 3, 1\} = \{3, 1\}$

General Algorithms

- *Take the Most Cookies Algorithm*: the monster takes as many cookies as possible for each move.

General Algorithms

- *Take the Most Cookies Algorithm*: the monster takes as many cookies as possible for each move.
- $\{20, 19, 14, 7, 4\}$

General Algorithms

- *Take the Most Cookies Algorithm*: the monster takes as many cookies as possible for each move.
- $\{20, 19, 14, 7, 4\}$
- $\xrightarrow[\text{Jar 1, Jar 2, Jar 3}]{-14} \{6, 5, 0, 7, 4\} = \{6, 5, 7, 4\}$

General Algorithms

- *Binary Algorithm*: the monster takes 2^k cookies from all jars that contain at least 2^k cookies for k as large as possible.

General Algorithms

- *Binary Algorithm*: the monster takes 2^k cookies from all jars that contain at least 2^k cookies for k as large as possible.
- $\{18, 16, 9, 8\}$

General Algorithms

- *Binary Algorithm*: the monster takes 2^k cookies from all jars that contain at least 2^k cookies for k as large as possible.
- $\{18, 16, 9, 8\}$
- $\xrightarrow[\text{Jar 1, Jar 2}]{-16} \{2, 0, 9, 8\} = \{2, 9, 8\}$

General Algorithms

- *Binary Algorithm*: the monster takes 2^k cookies from all jars that contain at least 2^k cookies for k as large as possible.
- $\{18, 16, 9, 8\}$
- $\xrightarrow[\text{Jar 1, Jar 2}]{-16} \{2, 0, 9, 8\} = \{2, 9, 8\}$
- None of these algorithms are optimal in all cases.

$CM(S)$

- Suppose S is the set of the numbers of the cookies in the jars.

$CM(S)$

- Suppose S is the set of the numbers of the cookies in the jars.
- The *Cookie Monster number of S*, $CM(S)$, is the minimum number of moves Cookie Monster must use to empty all of the jars in S.

$CM(S)$

- Suppose S is the set of the numbers of the cookies in the jars.
- The *Cookie Monster number of S*, $CM(S)$, is the minimum number of moves Cookie Monster must use to empty all of the jars in S.
- Suppose $CM(S) = n$ and Cookie Monster follows an optimal procedure.

$CM(S)$

- Suppose S is the set of the numbers of the cookies in the jars.
- The *Cookie Monster number of S*, $CM(S)$, is the minimum number of moves Cookie Monster must use to empty all of the jars in S.
- Suppose $CM(S) = n$ and Cookie Monster follows an optimal procedure.
- After he performs move n, all jars are empty.

$CM(S)$

- Suppose S is the set of the numbers of the cookies in the jars.
- The *Cookie Monster number of S*, $CM(S)$, is the minimum number of moves Cookie Monster must use to empty all of the jars in S.
- Suppose $CM(S) = n$ and Cookie Monster follows an optimal procedure.
- After he performs move n, all jars are empty.
- Therefore, each jar may be represented as the sum of some moves.

General Bounds for $CM(S)$

Theorem

For all $|S| = m$, $\lceil \log_2(m+1) \rceil \leq CM(S) \leq m$.

Interesting Sequences

- We now present our Cookie Monster with interesting sequences of cookies in his jars.

Interesting Sequences

- We now present our Cookie Monster with interesting sequences of cookies in his jars.
- First, we challenge our monster to empty a set of jars containing cookies in the Fibonacci sequence.

Interesting Sequences

- We now present our Cookie Monster with interesting sequences of cookies in his jars.
- First, we challenge our monster to empty a set of jars containing cookies in the Fibonacci sequence.
- Define the Fibonacci sequence as $F_0 = 0$, $F_1 = 1$, and $F_i = F_{i-2} + F_{i-1}$ for $i \geq 2$.

Interesting Sequences

- We now present our Cookie Monster with interesting sequences of cookies in his jars.
- First, we challenge our monster to empty a set of jars containing cookies in the Fibonacci sequence.
- Define the Fibonacci sequence as $F_0 = 0$, $F_1 = 1$, and $F_i = F_{i-2} + F_{i-1}$ for $i \geq 2$.
- A jar with 0 cookies and 2 jars containing 1 cookie are irrelevant, so our smallest jar will contain F_2 cookies.

Fibonacci and $CM(S)$

Theorem

When $S = \{F_2, \ldots, F_m\}$*, then* $CM(S) = \lceil \frac{m}{2} \rceil$*.*

N-nacci

- Each term starting with the n-th is the sum of the previous n terms.

N-nacci

- Each term starting with the *n*-th is the sum of the previous *n* terms.
- For example, in the 3-nacci sequence, otherwise known as Tribonacci, each term after the third is the sum of the previous three terms.

N-nacci

- Each term starting with the n-th is the sum of the previous n terms.
- For example, in the 3-nacci sequence, otherwise known as Tribonacci, each term after the third is the sum of the previous three terms.
- 0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149 . . .

N-nacci

- Each term starting with the n-th is the sum of the previous n terms.
- For example, in the 3-nacci sequence, otherwise known as Tribonacci, each term after the third is the sum of the previous three terms.
- 0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149 ...
- Tetranacci, Pentanacci similar

Tribonacci Example

- $\{1, 2, 4, 7, 13, 24, 44\}$

Tribonacci Example

- $\{1, 2, 4, 7, 13, 24, 44\}$
- $\xrightarrow[\text{Jar 6 , Jar 7}]{-24} \{1, 2, 4, 7, 13, 0, 20\} = \{1, 2, 4, 7, 13, 20\}$

Tribonacci Example

- $\{1, 2, 4, 7, 13, 24, 44\}$
- $\xrightarrow[\text{Jar 6 , Jar 7}]{-24} \{1, 2, 4, 7, 13, 0, 20\} = \{1, 2, 4, 7, 13, 20\}$
- $\xrightarrow[\text{Jar 5, Jar 6}]{-13} \{1, 2, 4, 7, 0, 7\} = \{1, 2, 4, 7\}$

Tribonacci Example

- $\{1, 2, 4, 7, 13, 24, 44\}$
- $\xrightarrow[\text{Jar 6 , Jar 7}]{-24} \{1, 2, 4, 7, 13, 0, 20\} = \{1, 2, 4, 7, 13, 20\}$
- $\xrightarrow[\text{Jar 5, Jar 6}]{-13} \{1, 2, 4, 7, 0, 7\} = \{1, 2, 4, 7\}$
- "Empty" 3 jars with 2 moves!

Tribonacci and $CM(S)$

Theorem

When $S = \{T_3, \ldots, T_m\}$*, then* $CM(S) = \lceil \frac{2m}{3} \rceil - 1$*.*

General Nacci $CM(S)$

Theorem

When Cookie Monster is presented with the first $m - (n - 1)$ *distinct n-nacci numbers, CM(S) =* $\lceil \frac{n-1}{n} m \rceil - (n - 2)$.

Beyond Nacci

- The monster wonders if he can extend his knowledge of nacci sequences to non-nacci ones.

Beyond Nacci

- The monster wonders if he can extend his knowledge of nacci sequences to non-nacci ones.
- Define a Super-n-nacci sequence as $S = \{k_1, \ldots, k_m\}$ where $k_{i+n} \geq k_{i+n-1} + \cdots + k_i$ for $i \geq 1$.

Beyond Nacci

- The monster wonders if he can extend his knowledge of nacci sequences to non-nacci ones.
- Define a Super-n-nacci sequence as $S = \{k_1, \ldots, k_m\}$ where $k_{i+n} \geq k_{i+n-1} + \cdots + k_i$ for $i \geq 1$.
- Cookie Monster suspects that since he already knows how to consume the nacci sequences, he might be able to bound $CM(S)$ for Super naccis.

Super Nacci

Theorem

For Super-n-nacci sequences with m terms, $CM(S) \geq \lceil \frac{(n-1)m}{n} \rceil$.

Bounds Depending on Growth of Sequence

- The *n*-nacci sequences all give monic recursive equations, $x^n = x^{n-1} + x^{n-2} + \cdots + x^1 + 1$.

Bounds Depending on Growth of Sequence

- The n-nacci sequences all give monic recursive equations, $x^n = x^{n-1} + x^{n-2} + \cdots + x^1 + 1$.
- Therefore, any n-nacci sequence approximates a geometric sequence, specifically $\alpha r, \alpha r^2, \alpha r^3, \ldots$ where r is a real root of the characteristic polynomial and α is some real number.

Bounds Depending on Growth of Sequence

- The n-nacci sequences all give monic recursive equations, $x^n = x^{n-1} + x^{n-2} + \cdots + x^1 + 1$.
- Therefore, any n-nacci sequence approximates a geometric sequence, specifically $\alpha r, \alpha r^2, \alpha r^3, \ldots$ where r is a real root of the characteristic polynomial and α is some real number.
- Cookie Monster wonders if there is any relationship between r and $\frac{n-1}{n}$, the coefficient for $CM(S)$ of n-nacci.

Bounds Depending on Growth of Sequence

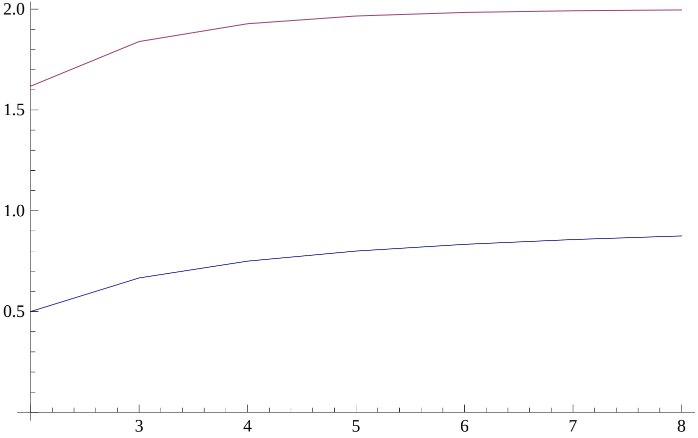


Figure: Real root (red) approaches 2 as fraction (blue) approaches 1.

Bounds Depending on Growth of Sequence

Theorem

The real root of $x^n = x^{n-1} + x^{n-2} + \cdots x + 1$ *may be approximated as* $2 - \epsilon$ *where* $\epsilon = \frac{1}{2^n - 1}$.

Bounds Depending on Growth of Sequence

Theorem

The real root of $x^n = x^{n-1} + x^{n-2} + \cdots x + 1$ *may be approximated as* $2 - \epsilon$ *where* $\epsilon = \frac{1}{2^n - 1}$.

- Thus, $n \approx \log_2(1 + \frac{1}{2-r})$, so $\frac{n-1}{n} = 1 - \frac{1}{n} \approx 1 - \frac{1}{\log_2(1+\frac{1}{2-r})}$.

Bounds Depending on Growth of Sequence

Theorem

The real root of $x^n = x^{n-1} + x^{n-2} + \cdots x + 1$ *may be approximated as* $2 - \epsilon$ *where* $\epsilon = \frac{1}{2^n - 1}$.

- Thus, $n \approx \log_2(1 + \frac{1}{2-r})$, so $\frac{n-1}{n} = 1 - \frac{1}{n} \approx 1 - \frac{1}{\log_2(1+\frac{1}{2-r})}$.
- Therefore, for large n, the Cookie Monster coefficient approximates a function of r.

Bounds Depending on Growth of Sequence

Conjecture

There exist m jars in a recursive sequence with characteristic equation of the form $x^n = x^{n-k_1} + x^{n-k_2} + \cdots x^{n-k_m}$ *such that* $CM(S) = \lceil qm \rceil$ *where q is any rational number between 0 and 1.*

Future Research

- $CM(S) = n$ characterization
- Explore more interesting sequences with regard to $CM(S)$
- Is computing $CM(S)$ an NP-hard problem?
- Introduce more monsters, more dimensions of cookies, or a game to play with the cookies

Acknowledgements

- Tanya Khovanova
- PRIMES
- My parents

Thanks for listening!