RANDOMLY PICKING THE CORRECT NUMBER SOMEHOW? AN INTRODUCTION INTO THE PROBABILISTIC METHOD

JASMINE TANG

Abstract. Consider the probabilistic method pioneered by the late Paul Erdős. This expository paper then gives a gentler introduction to the topic of Games and attempts to explain the use of the probabilistic method in this field to the reader.

Preface

Since the author of this expository paper likes writing a lot, especially emotion-inducing writings, this paper, of course, totally, no doubts, ought to, has to begin with two thank-you notes. The first one is for my mentor Anna Brosowsky, and the second one is for Stephen McKean and Julius Frank, the program organizers of this math directed reading program.

Thank you. I would like to express my most sincere appreciation and gratitude to Anna Brosowsky for being an amazing mentor. Your flexibility and understanding allowed me to explore different topics in Math and ultimately select my favorite research topic. You gave me autonomy and supported my decisions wholeheartedly. I was able to learn math from different levels, as well as cool proof techniques from the material you provided. Your generous commitment to volunteer your time every week to teach me new math was invaluable. You even stayed a few minutes longer to make sure that I finally understood an important concept. I can't thank you enough for all you have done for me. Your patience, guidance, and friendship have been a tremendous blessing, and I am extremely grateful.

And thank you. Hi there Stephen! Hi there Julius! It's Jasmine here! Thank you for organizing this online math directed reading program. Seriously, thank you thank you thank you. Your trust and acceptance of me into the program has completely changed my life. As a community college student whose institution does not have much resources to begin with, let alone for students with a research interest blended between computer science, math and physics, your program has been a wonderful journey. Thanks to the research experience I gained through your program, I have been accepted into a full-time research internship at Fermilab, where I will be simulating and analyzing accelerators' beams this Spring. Your program opened another door for students like me, and I am truly thankful. Your commitment to helping students reach their potential is inspiring, and I am forever grateful for all you have done. I sincerely thank you again for doing this.

Date: November 2022.

0.1. **Who is this paper for?** This expository paper is geared toward freshman/sophomore students of Computer Science and Math majors. Readers should have taken a proof-style Math/Computer Science class before. It also assumes the readers are familiar with topics in Single Variable Calculus 1-2, Introductory Probability, and Introductory Algorithms. Specifically, it assumes the following from readers.

- Precalculus/Calculus 1/Calculus 2 topics (based on different curriculums): Basics knowledge of Series.
- Computer Science topics: Binary Searching.
- Probability Topics: Concepts of Probability, Expected Values, and Basics Combinatorics knowledge.
- Misc: Basic proof techniques, set notation.

1. Introduction

Throughout the notes, I use formulas and theorems from Alon & Spencer's "The Probabilistic Method" [AS08].

1.1. **What and Why.** "What do we mean by GAME?" In this case, we mean a game by a competition between two people, each trying to win the designated objective and having the property of a perfect information game. ((INSERT PERFECT INFORMATION CITATION))

"Why these games?" The games in this paper are accessible to the audience and thus are good for showing the attractiveness and complexity of the field of game theory.

"Why the probabilistic method?" The probabilistic method is designed with the purpose of combating the exponential growth of cases in combinatoric problems. When using the method, we are not listing things case by case anymore but instead employ randomness and probability to argue that the wanted event has positive probability, thus deterministically proving the existence of our result. By studying the probabilistic method and examining the proofs in this expository paper, readers can transition from a rigid proof style to a more varied and diverse style of proof.

Plus, per Twoples' rules, I get to write this expository paper and I love it hehe :)

1.2. **Prior Reading.** Readers are encouraged to read up Section 1.1 and Section 2.1 for basic coverage. Here, we introduce only important concepts to the following games from Section 2.1: Linearity of Expectation and Zermelo's Theorem

Theorem 1.1. *Let $X_1, ..., X_n$ be random variables, and let X be the linear combination of those random variables, i.e $X = c_1X_1 + ...c_nX_n$.*

Then, the linearity of expectation states that the expected value of X also follows the same linear combination principle:

$$\mathbb{E}[X] = c_1\mathbb{E}[X_1] + c_2\mathbb{E}[X_2] + ... + c_n\mathbb{E}[X_n]$$

Proof. There is no proof, as it is beyond the scope of this paper. □

2. The Liar Game

The following section comes from Chapter 15.2's material of Alon & Spencer [AS08].

Game. Suppose Lucy are given a range of numbers from 1 to n by our Jasmine, and we have to deduce the correct number x through a series of maximum q queries by the question: "Is x in S," where S is one of the subsets from 1 to n. Jasmine is rather a sly person, so she would lie at most k times to the queries. The integral question is, "For what n, k, q can we guess the correct number every time?"

Let us look at an example of this game before moving on.

Example 2.1. Lucy is given a range $[1, 4]$ by Jasmine. She can give 2 queries to Jasmine, and Jasmine cannot lie to any query. Then $n = 4, q = 2, k = 0$. Suppose $x = 3$. A possible query from Lucy can be "Is x in the subset $S = \{1, 2\}$" and a possible response from Jasmine can be "Nuh uh, try again."

The above example gives rise to an interesting observation.

Theorem 2.2. *When $k = 0$, Lucy has a winning strategy if and only if $n \leq 2^q$.*

Proof. The result comes from the idea of binary searching that readers might be familiar with in an introductory class on algorithms. Each time when Lucy queries Jasmine, she can split the range of numbers into two smaller ranges of equal size and ask if x is in the lower range or not, effectively cutting the range by half for every query.

To see the opposite direction (why Lucy would lose if $n > 2^q$), we again use the idea of binary search that every query cuts the range by half. Thus, as we successively query q rounds, we effectively divide the range by 2^q, thus $\frac{n}{2^q} > 1$. We can see after q rounds, there is (sometimes) still more than one possibility for Lucy to choose from. Thus, Lucy cannot always win. ∎

Example 2.3. Let us apply the strategy mentioned above to Example 2.1.

Supposed Jasmine picked $x = 3$ again.

Then Lucy can apply the strategy to always systematically query the first half of the set [1]: $\{1, 2\}$. Jasmine says No. So Lucy can cut down the size of n by half, from 4 to 2. Lucy then again queries the first half of the remaining set $\{3, 4\}$: 3

Jasmine will have to say "Yes", she reveals the answer herself. Thus, Lucy wins.

We managed to find an explicit strategy deterministically for $k = 0$, without any randomization or probability. But is there any way we can deduce the result for Lucy when n, q, k is arbitrary? Even though we cannot even imagine the complexity when n, q, k gets larger, we can rely on the probabilistic method to deduce the existence of a winning strategy for Lucy.

At some point, we also need to consider Jasmine in the equation. After all, humans are complex creatures with intelligence capable of adapting to new situations and manipulating them using knowledge. It would do Jasmine no justice, just allowing her to answer only "Yes" or "No" to Lucy's mundane queries. Let her make use of the information around her! And give her a chance to actually compete in this game too!

Game. Let us reframe the game into a two-sided game where Jasmine is also playing it to turn this into a perfect information game. Jasmine will not pick an x in advance but rather use the information given by Lucy (through queries) to always answer consistently with one x. Doing this, she can refuse to pick any x,

[1](in this case, the set is all natural numbers from 1 to 4)

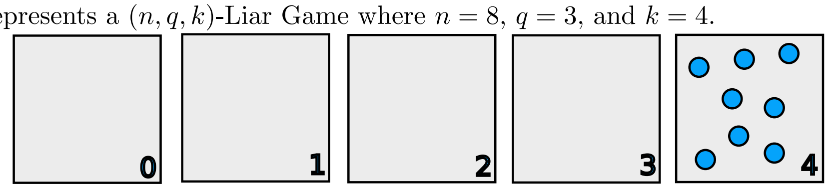
but her valid “Yes” and “No” answers ensure her a choice to pick from if Lucy ever questions the integrity of the game.

At the end of the game, if Jasmine’s answers are consistent with more than one x, then she has won. In other words, as long as Lucy is not able to deduce with 100% which x to pick, Jasmine will win, and Lucy will lose. If Lucy is able to deduce it, then Lucy will win, and Jasmine will lose. To see why, consider when Jasmine’s last answer is consistent with more than one x, then Lucy would not be able to deduce which one to pick.

We also introduce the Chip-Liar Game and show that it is equivalent to the (n,q,k) Liar Game.

Game (The Chip-Liar Game)**.** Imagine a board of positions: $0, 1, ..., k$, and in the beginning, all of n chips are put at position k. A total of q rounds are played. Each round, Lucy selects a set of chips S from the set N of n chips and brings it to Jasmine. Jasmine then can either move the set S or the complement S' to the left. By moving the set, we mean moving the position of each chip in the set to the left. For example, if a chip has position i, we move it to $i-1$. If a chip has a position less than 0, it means that it is wiped from the board. At the end of the rounds, Jasmine wins if there is more than one chip remaining on the board.

Example 2.4. Suppose Lucy can ask only 3 questions, then the picture below represents a (n, q, k)-Liar Game where $n = 8$, $q = 3$, and $k = 4$.



Theorem 2.5. *The Chip-Liar Game is equivalent to the Liar Game*

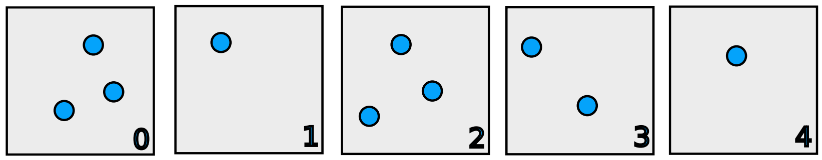
Proof. The act of selecting a set S stands for querying the question: “Is x in S?” Moving a set S is the same as Lucy and Jasmine’s agreeing to get rid of the set, i.e Jasmine saying “No, x is not in the set” and Lucy trusts her. If Jasmine answers “No” to the query “Is x in S?”, they have to move set S to the left. If Jasmine answers “Yes” to “Is x in S?”, then they both know that x is not in S', thus they has to move S' to the left.

A chip i that has position j means that the question “Is i the number” has (potentially) received $k - j$ lies. To see why, consider the case where k=2, then Lucy would have to ask the same question 3 times before she is 100% sure if x is really in S or not. □

Before we figure the answer to determine the result of the n, q, k Liar Game, we relax on the condition of the Chip-Liar Game to develop a theorem that is applicable to our Liar Game.

We relax the Chip-Liar Game to the $(x_0, ..., x_k), q$-Chip-Liar Game where initially, the position i has x_i chips and $x_0 + x_1 + ... + x_k = n$. We then realize that when $x_k = n$ and other $x_i = 0$, the $(x_0, ..., x_k), q$-Chip-Liar Game is equivalent to the Chip-Liar Game, which in turn is equivalent to the n, q, k Liar Game.

Example 2.6. Suppose Lucy can ask at most 3 questions, then picture below represents a $(3, 1, 3, 2, 1)$-3-Chip-Liar Game



Notice that in the case of $k = 0$, we don't have to worry about Jasmine's lying. However, when k is arbitrary, looking from Lucy's perspective, we can see that there are some randomness involve.

The move then becomes trying to prove that the probability of a wanted event is always strictly larger than 0 or the considered expected value reaches a certain threshold in every case. By doing this, we rely on randomness to deduce a deterministic answer, thus, the essence of the probabilistic method.

But, let us define a probability formula that will help us with the probabilistic method first.

Fact 2.7. *"Define $B(q, j)$ as the probability that in q flips of a fair coin, there are at most j heads." [AS08, p. 258], then*

$$B(q, j) = 2^{-q} \sum_{i=0}^{j} \binom{q}{i}$$

Proof. $\binom{q}{i}$ is the number of combination of flipping a coin q times that has i heads and $q - i$ tails. Each time flipping producing either heads (or tails) occurs with $\frac{1}{2}$ probability. Thus, the probability of flipping a coin q times and have exactly i heads is $(\frac{1}{2})^q \binom{q}{i} = 2^{-q}\binom{q}{i}$. Since the case of at most j heads consists of individual cases of i heads: $i = 1, 2, 3, 4, ..., j$:

$$\begin{aligned} B(q, j) &= \sum_{i=0}^{j} 2^{-q} \binom{q}{i} \\ &= 2^{-q} \sum_{i=0}^{j} \binom{q}{i} \end{aligned}$$

□

Then we can introduce the theorem to our question

Theorem 2.8. *If*

$$\sum_{i=0}^{k} x_i B(q, i) > 1$$

then Jasmine wins the $(x_0, ..., x_k), q$-Chip-Liar Game

Proof. Focusing on Jasmine's perspective and let her play randomly in the way she moves the chips. More specifically, after Lucy has selected a set S and queries it, Jasmine flips a coin to answer Lucy. Jasmine moves S' to the left if the coin is head, and she moves S to the left if the coin is tail. For each chip c, let I_c be the boolean random value for c remaining on board. That is if c remains on the board, then $I_c = 1$ and if c is no longer on the board, $I_c = 0$. Then the total number of chips remaining on the board is just the boolean value I_c of each chip combined. We call this $X = \sum I_c$.

Considering the gist of the probabilistic method, if we can get the expected value (of the total number of chips X) to be larger than 1, then we can proven that Jasmine would win the game.

Every round that Lucy and Jasmine play, a chip c has probability of $\frac{1}{2}$ to move to the left. If c starts at position i, then c remains on the board if and only if in q rounds, it has only been moved to the left at most i times. For example, if c starts at position 2, it can only remain if it has been moved at most 2 time. Then the probability of a chip c having been moved to the left at most i times is $B(q, i)$. We then aim to compute the expected value of I_c, which is $\mathbb{E}[I_c]$.

$$\begin{aligned}\mathbb{E}[I_c] &= \textbf{Sum of each value of } I_c \textbf{ times the probability of } I_c \textbf{ equaling the value}\\ &= 1 * \text{Probability of } c \text{ remaining on the board}\\ &\quad + 0 * \text{Probability of } c \text{ not remaining on the board}\\ &= 1 * B(q,i) + 0 * (1 - B(q,i))\\ &= 1 * B(q,i) + 0\\ &= B(q,i)\end{aligned}$$

The reason we want to figure out the boolean expected value of a chip is because of the linearity property of expectation: "The expected value of sum of random variables is equal to the sum of expected value of each random variables"

$$\begin{aligned}\mathbb{E}[X] &= \mathbb{E}[\sum I_c]\\ &= \sum \mathbb{E}[I_c]\\ &= \sum_{i=0}^{k} x_i B(q,i)\end{aligned}$$

Since there are x_i chips in the x_i position, each with the expected value of $\mathbb{E}[I_c] = (q, i)$, we get the answer $\sum_{i=0}^{k} x_i B(q, i)$.

By the initial assumption of the proof's statement, $\mathbb{E}[X] > 1$. The average of X is larger than 1, which means at least one possible value of $X > 1$ has to occur, which means $X > 1$ has to occur with positive probability. Since X, the total number of chips, has positive probability to be larger than 1, Carole has positive probability to win.

Then, "No strategy of Lucy allows her to always win. But this is a perfect information game with no draws so someone has a perfect strategy that always win. That someone isn't Lucy, so it must be Jasmine"[2] [AS08] [[TODO for Anna: continue to try to find a citation for this pure strategy result]] □

Corollary 2.9. *If*

$$n > \frac{2^q}{\sum_{i=0}^{k} \binom{q}{i}}$$

then Jasmine wins the (n,q,k)- Liar Game.

[2] I replaced Paul with Lucy and Carole with Jasmine.

Proof. The (n,q,k)- Liar Game is the special case of the $(x_0, ..., x_k), q$-Chip-Liar Game when $x_k = n$ and other $x_i = 0$

Then continuing the discussion of $E[X]$ from the previous proof

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{i=0}^{k} x_i B(q,i) \\
&= x_k * B(q,k) + \sum_{i=0}^{k-1} x_i B(q,i) \\
&= x_k * B(q,k) + \sum_{i=0}^{k-1} 0 * B(q,i) \\
&= x_k * B(q,k) + 0 \\
&= x_k * B(q,k) \\
&= n * B(q,k) \\
&= n * \sum_{i=0}^{j} 2^{-q} \binom{q}{i}
\end{aligned}$$

We perform algebraic manipulation on the proof's assumption and show that the assumption implies that the expected value of the total number of chips, $E[X]$, is larger than 1:

$$\begin{aligned}
n &> \frac{2^q}{\sum_{i=0} k\binom{q}{i}} \\
n * 2^{-q} &> \frac{2^q}{\sum_{i=0} k\binom{q}{i}} * 2^{-q} \\
n * 2^{-q} &> \frac{1}{\sum_{i=0}^{k} \binom{q}{i}} \\
n * 2^{-q} * \sum_{i=0}^{k} \binom{q}{i} &> \frac{1}{\sum_{i=0}^{k} \binom{q}{i}} \sum_{i=0}^{k} \binom{q}{i} \\
n * 2^{-q} * \sum_{i=0}^{k} \binom{q}{i} &> 1 \\
n * \sum_{i=0}^{k} 2^{-q} \binom{q}{i} &> 1 \\
\mathbb{E}[X] &> 1
\end{aligned}$$

The rest of the proof follows the same as the previous proof. □

Take notice that all the examples and theorems we have done do not mean that when we achieve the required condition(s) of the theorem, we always win by doing random things. For example, with theorem 2.2 and example 2.1, Lucy can query 2 times with $S = \{1\}, \{2\}$ and would lose if $x = 3$ or 4. They only guarantee that when we achieve the required condition(s), there exists a strategy so that we always win.

Then what the players want to know (and us too!) is an explicit strategy, and the transition "from a probabilistic existence proof to an explicit construction is called derandomization... Here we can give an explicit strategy."[AS08, p. 259]

Theorem 2.10. *Explicit strategy: Suppose there are l moves remaining in the game and y_i chips at position i, define the weight $W = \sum_i y_i B(l, i)$. Suppose also from theorem 2.8 that the weight W is larger than 1. Notice a similarity from Theorem 2.8 that the weight W is just the expected value of Y, where Y is the random variable denoting the total number of chips at the end if Jasmine plays randomly. If Jasmine can somehow make the weight W be larger than 1 at the end of the game, she would win.*

Proof. Let W^y be the new expected value if Jasmine says "Yes" to the query "Is x in S" and move S' to the left.

Let W^n be the new expected value if Jasmine says "No" to the query "Is x in S" and move S to the left.

As Jasmine can only say "Yes" or "No" for an answer, and suppose she were to play randomly (to obey the expected value of Y, i.e W), then $W = \frac{1}{2} * W^y + \frac{1}{2} * W^n = \frac{1}{2}(W^y + W^n)$. This can be explained by Jasmine's first flipping the coin and then deciding whether she should answer "Yes" or "No".

Jasmine's plan for this game will revolves around maximizing the weight of the chips. Let us show exhaustively how this strategy works. Suppose the weight $W > 1$, then

$$W = \frac{1}{2}(W^y + W^n) > 1$$
$$W^y + W^n > 2$$

We consider 4 cases of W^y and W^n either above or below 1:

- If $W^y \leq 1$ and $W^n \leq 1$, this case never happens since $W^y + W^n \leq 2 \not> 2$, thus contradicting the assumption of the theorem.
- If $W^n \leq 1$, then $W^y > 2 - W^n > 1$. Jasmine will choose to answer "Yes" in this case to maximize the weight and keep it above 1.
- If $W^y \leq 1$, then $W^n > 2 - W^y > 1$. Jasmine will choose to answer "No" in this case to maximize the weight and keep it above 1.
- If $W^y > 1$ and $W^n > 1$, then Jasmine will choose to answer "Yes" or "No" based on the maximum between W^y and W^n, also keeping it above 1. She particularly randomly answers "Yes" or "No" if the two weights are equal.

Since we assume initially that the weight W is larger than 1 and Jasmine makes sure that the updated weight will always be larger than 1, at the end of the game, the weight will be larger than one. As the weight at the end of the game will just be the number of chips remaining, this means the number of chips has to be at least two. Since the game ends with two or more chips, Jasmine wins the game. □

3. The Tenure Game

Let us play an unrealistic political game in academia, the Tenure game, with the couple Lucy and Jasmine.

Game. Lucy wants to promote the faculties to tenure but Jasmine, her enemy, is in the admission council. There are k pre-tenure levels: $\{1, ...k\}$ with level 1 be the highest pre-tenure level and level 0 be the tenure level. Let each applicant be represented by a chip. As the pool of application has different tenure levels, this promoting game to tenurity is a $(x_1, ..., x_k)$-Tenure Game begins with x_i chips level for $1 <= i <= k$ and no chips on level zero. Each year Lucy gives a set S of chips to Jasmine with $|S| > 0$ and Jasmine can either promote S and fire S', or promote S' and fire S.

Promote as in moving from i to $i - 1$ and fire as in completely removing the chips from the chip board.

If a chip reaches level 0, then Lucy has successfully promote a faculty to tenure and win the game. If none of the chips reaches level 0, i.e no faculty becomes tenure, then Jasmine will win the game.

By assuming that every query S has a size larger than 0 and enforcing the promotion and removal of chips at every level, we guarantee that at the end of the k-th round, we have at least a tenured faculty or no faculty at all. In fact, why don't we introduce a proof for this remark.

Theorem 3.1. *At the end of the $(x_1, ..., x_k)$-Tenure Game, which is the k-th round, we end up with either at least a tenured faculty or no faculty at all.*

Proof. We proceed to do induction on the number of rounds.

Base case: k = 1. If there is only 1 round, then by Lucy's picking and querying a set S, Jasmine will either have to promote S and fire S' or vice versa. Without loss of generality, let Jasmine promote S. If S is non-empty, we end up with at least a tenured faculty. If S is empty, then we have no faculty at all.

Suppose the theorem is true up to k rounds. We then consider the case $k + 1$.

Case k+1: Consider the first round of $k + 1$ rounds where Lucy queries a set S. Then after the first round, there are no more chips at position $k + 1$. To see why, let us consider a single chip c at position $k + 1$. That chip then has to either be in the promotion set or in the fire set. Thus it has to move to position k or exit the game, meaning no more chips at position $k + 1$.

We then remark that after the first round of case $k+1$, we essentially are instead playing the first game of $(x_1, ..., x_k)$-Tenure Game (since there are only k rounds left, with all the chips between x_1 and x_k. Since the theorem is true up to the case k, at the end of the game for case $k + 1$, we also end up with at least a tenured faculty or no faculty at all.

Thus, the induction is complete. □

Theorem 3.2. *If $\sum x_i 2^{-i} < 1$ then Jasmine wins the $(x_1, ..., x_k)$-Tenure Game*

Proof. As usual, we want to utilize the probabilistic method. Let Jasmine play randomly. Every round of promotion, after Lucy queries a set S, Jasmine flips a coin to make a decision. If it goes up head, Jasmine promotes S' and fire S. If it goes up tail, Jasmine promotes S and fire S'. For each chip c in level i, let I_c be the boolean random variable for c reaching the level 0. If $I_c = 1$, c is tenured and vice versa. Then the only way for $I_c = 1$ is every coin flip in the sequence of i coin flips has to lead to c moving to the left.

Then the expected value of I_c,

$$\begin{aligned}\mathbb{E}[I_c] &= 1 * 2^{-i} + 0 * (1 - 2^{-i}) \\ &= 2^{-i}\end{aligned}$$

Let $X = \sum I_c$. Then by linearity of expectation,

$$\begin{aligned}\mathbb{E}[X] &= \mathbb{E}[\sum I_c] \\ &= \sum \mathbb{E}[I_c] \\ &= \sum_{i=1}^{k} x_i * 2^{-i}\end{aligned}$$

Since we assume that $\mathbb{E}[X] < 1$, X has a positive probability to be less than 1, which is 0. Since X represents the sum of all chips c, $X = 0$ occurs only if every $c = 0$, which in turn means the event that no faculty chosen by Lucy is tenured occurs with positive probability. This means that Jasmine win with positive probability.

"No strategy of Paul allows him to always win. But this is a perfect information game with no draws so someone has a perfect strategy that always wins. That someone isn't Paul, so it must be Carole" [AS08, p. 261]. □

We also want to have an explicit strategy for the tenure game.

Theorem 3.3. *If $\sum x_i 2^{-i} < 1$, Jasmine can devise a strategy to always win the game.*

Proof. Let $W = \sum_i y_i 2^{-i}$ be the weight of the y_i chips on position i. The weight stands for the expected value of Y, where Y is the number of chips at the end if Jasmine decides to play randomly. Let W^y and W^n be the new weights if Jasmine says "Yes" and "No" respective as an answer. Then just with the same logic as the Liar Game, $W = \frac{1}{2}(W^y + W^n)$

With this explicit strategy, Jasmine always will always want to minimize the weight. Assuming the weight (i.e expected value) to be less than 1, a similar argument can be made [3] just like theorem 2.10 to show that Jasmine can always update the weight so that it is always less than 1. Since the weight at the end of the game, which is the number of chips, is less than 1, there is zero chips at the end of the game. This results in Jasmine winning the game as there is no faculty becoming tenure. □

"Wait a minute!" - a group of Lucifers [4] exclaims.

"I want Lucy to win!" - they angrily argues. "A person so beautiful, kind and smart deserves to have her own winning play in this game. We are tired of Jasmine always winning ever since you promoted her to an even playing field."

And thus, the author of this expository paper rolls her eyes, lets out a subtle "hmmphh" and are forced to introduce a lemma and a theorem that reveals Lucy's winning play.

((Sometimes I see very big whitespace between theorems and math derivations))

Lemma 3.4. *If a set of chips has weight at least one it may be split into two parts, each of weight at least $\frac{1}{2}$.*

[3] Readers can search up the Pigeonhole principle to avoid arguing exhaustively

[4] Fans of Lucy

Proof. Suppose a set of chips has a weight of at least one. Then there has to be at least two chips at some position i. As a counter-example, suppose all position only have 1 chip. Then

$$
\begin{aligned}
W &= \sum_{i=1}^{k} y_i 2^{-i} \\
&= \sum_{i=1}^{k} 2^{-i} \\
&= \frac{1}{2} + \frac{1}{4} + ... + \frac{1}{2^k}
\end{aligned}
$$

The sum above results from Zeno's Paradox, with its finite sum (the weight W) always less than 1.

We proceed to do induction on the number of chips $n = \sum_{i=1}^{k} x_i$.

Base case: Let $n = 1$, then on the first position $i = 1$, there are 2 chips with $W = 2 * \frac{1}{2} = 1$. We can split them into two individual set on the same position $i = 1$, consisting of only 1 chip, then the weight of each chip is $W = 1 * \frac{1}{2} = \frac{1}{2}$

Supposed the theorem works for up to n number of chips. Then we would want to prove that the theorem works for $n + 1$ number of chips also.

Case $n + 1$: Let the set S be the set of all $n + 1$ chips. Also let the set S' be of size of n, but with the same weight as the set S, with the only difference is that at position i where there are two chips, we compress the chips into 1 superchip at position $i - 1$. We show the concept mathematically as follows. Let W' be the weight of the set S', and $z_t = y_t$ be the number of chips at position t of S', with $z_{i-1} = y_{i-1} + 1$ and $z_i = y_i - 2$ due to compression of the two chips at level i. Then

$$W = \sum_{t=1}^{K} y_t 2^t \tag{1}$$

$$= \sum_{t=1}^{i} y_t 2^t + \sum_{t=i+1}^{k} y_t 2^t \tag{2}$$

$$= \sum_{t=1}^{i-2} y_t 2^t + \frac{y_{i-1}}{2^{i-1}} + \frac{y_i}{2^i} + \sum_{t=i+1}^{k} y_t 2^t \tag{3}$$

$$= \sum_{t=1}^{i-2} y_t 2^t + \frac{y_{i-1}}{2^{i-1}} + \frac{2 + y_i - 2}{2^i} + \sum_{t=i+1}^{k} y_t 2^t \tag{4}$$

$$= \sum_{t=1}^{i-2} y_t 2^t + \frac{y_{i-1}}{2^{i-1}} + \frac{2}{2^i} + \frac{y_i - 2}{2^i} + \sum_{t=i+1}^{k} y_t 2^t \tag{5}$$

$$= \sum_{t=1}^{i-2} y_t 2^t + \frac{y_{i-1}}{2^{i-1}} + \frac{1}{2^{i-1}} + \frac{y_i - 2}{2^i} + \sum_{t=i+1}^{k} y_t 2^t \tag{6}$$

$$= \sum_{t=1}^{i-2} y_t 2^t + \frac{y_{i-1} + 1}{2^{i-1}} + \frac{y_i - 2}{2^i} + \sum_{t=i+1}^{k} y_t 2^t \tag{7}$$

$$= \sum_{t=1}^{i-2} z_t 2^t + \frac{z_{i-1}}{2^{i-1}} + \frac{z_i}{2^i} + \sum_{t=i+1}^{k} z_t 2^t \tag{8}$$

$$= \sum_{t=1}^{i} z_t 2^t + \sum_{t=i+1}^{k} z_t 2^t \tag{9}$$

$$= \sum_{t=1}^{k} z_t 2^t \tag{10}$$

$$= W' \tag{11}$$

Now S' possesses two characteristics: it has only n chips but has the weight of the set S. We have transformed the set S with $n+1$ chips into an n chips set S' with the same weight.

By assumption of the induction that the theorem works for case n (which refers S', which again refers to S), we have proven that the theorem also works for case $n+1$.

To use the decomposition for S' to build back a composition of S to finally finish the induction, we consider the following scenario: As we have successfully transform S to S' and split S' into two sets T and $T' = S' \setminus T$ so that $S' = T + T'$, then surely the superchip that we initially "create" in S' will either be in T or T'. We can then expand the superchip back into two normal chips. Thus the amount of chips return back to $n+1$ and we have our set S back. The transformation can be mathematically shown by backtracking from equation 9 to equation 1

This concludes the induction, therefore, the proof.

□

Theorem 3.5. *If* $W = \sum x_i 2^{-i} \geq 1$ *then Lucy wins the* $(x_1, ..., x_k)$*-Tenure Game*

Proof. Since the intial weight is at least 1, Lucy can use lemma 3.4 to split the total set S into two smaller sets that she is going to query: T and $T' = S \setminus T$, each with weight larger or equal to $\frac{1}{2}$.

As Lucy queries either set T or T', Jasmine will have either choice of promoting T and firing T', or promoting T' or firing T.

Without loss of generality, suppose Jasmine promote T, then each set of chips y_i at position i [5]in T gets pushed to the position $i-1$ to the left. Let $W(T_{\text{before}}) = \sum y_i 2^{-i} \geq \frac{1}{2}$ and $W(T_{\text{after}}) = \sum y_i 2^{-(i-1)}$ to respectively be the weight of the set T before and after the promotion. Then

$$\begin{aligned} W(T_{\text{after}}) &= \sum y_i 2^{-(i-1)} \\ &= \sum y_i 2^{-i+1} \\ &= \sum y_i 2^{-i} * 2 \\ &= 2 \sum y_i 2^{-i} \\ &= 2 * W(T_{\text{before}}) \\ &\geq 2 * \frac{1}{2} \\ &\geq 1 \end{aligned}$$

We have shown algebraically that in the process of promoting a set T or T', we end up doubling the set's weight, thus still keeping the new weight above or equal to 1 throughout the whole game. As the game goes on and ends without the weight going below one, this means that at least one chip must be remaining at the end of the game. Since we have shown the game can only end with either all chips being removed or at least a chip reaching level 0, a chip remaining at the end of the game translate to the same chip being at level 0. Thus Lucy successfully promotes her tenured faculty and wins the $(x_1, ..., x_k)$-Tenure Game

□

4. Implementations

Readers interested in implementing these ideas can visit this link to see my primitive implementations in Python's Jupyter Notebook.

Additionally, they can download the notebook and play around with it too!

References

[AS08] Noga Alon and Joel H Spencer. *The probabilistic method.* en. 3rd ed. Wiley Series in Discrete Mathematics and Optimization. Hoboken, NJ: Wiley-Blackwell, Aug. 2008.

Email address: `tanghocle456@gmail.com`

[5]we don't have to consider the case $i = 0$ because if there is a chip at level 0, Lucy has already won