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.
- A weekend's worth of time.

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 paper, we mean a game by a competition between two people who take turns, each trying to win the designated objective and having the property of a perfect information game [1]. In trying to win the game, they are aware that the choice that they make, as well as the choice that their opponent makes, can affect the overall outcome of the game. *"Why these games?"* The games in this paper are accessible to the target 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. **Background.** Readers are encouraged to read Section 1.1 and Section 2.1 of [AS08] for basic coverage. Here, we introduce only important concepts to the following games from Section 2.1: Linearity of Expectation, as well as sequential games, perfection information, and Zermelo's Theorem. We will not give proof of either of these as they are beyond the scope of this paper. However, a corollary is given with proof to accompany readers throughout the paper.

Theorem 1.1 (Linearity of Expectation)**.** *Let $X_1, \dots, X_n$ be random variables, and let X be a linear combination of those random variables, i.e., $X = c_1X_1 + \cdots + c_nX_n$.*

[1] which will be defined below.

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

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

In the context of this paper only, we provide some definitions in the field of game theory.

Definition 1.2 (Sequential game)**.** A sequential game is where one player makes their first move, followed by the other player making their move, knowing the first player's choice.

Definition 1.3 (Perfect information; see §6.4.1 in [vNM07])**.** A sequential game has the "perfect information" characteristic when both players know each other's previous moves and the initial condition of the game they are playing.

Example 1.4. Games such as tic-tac-toe, chess, and go are perfect information sequential games.

Theorem 1.5 (Zermelo's Theorem; see Ch. 15 and Thm 15:D of [vNM07])**.** *In a two-player perfect information game with no draws, one of the players has a strategy that will guarantee victory, regardless of the choices of the other player. In other words, exactly one of the following two scenarios occurs:*

- *Player 1 posses a strategy with which she "wins", irrespective of what player 2 does.*
- *Player 2 possesses a strategy with which she "wins", irrespective of what player 1 does.*

Per the title of the paper, we are sure to employ randomization to analyze these sequential perfect information games. Therefore, it is only natural to streamline this process for the readers by introducing a corollary suitable to the probabilistic method.

Corollary 1.6. *In a two-player perfect information game with no draws, if one of the players has a strategy with randomization which allows her to win with positive probability, then in fact, this player has a strategy* without *randomization with which she always wins.*

Proof. Suppose there are two players in a two-player perfect information game with no draws.

Without loss of generality, let's say Jasmine is the player who has a randomized strategy with which she wins with positive probability and let's call her opponent Lucy. By the previous theorem, one of these players has a winning strategy.

We proceed with a proof of contradiction.

Suppose Lucy is the winner instead. But then, since Lucy's strategy guarantees victory, Lucy would *always* win, and Jasmine would *always* lose. This means she wins with probability 0. This is a contradiction since we assume that Jasmine has a randomized strategy that can win with positive probability.

The game has no draws, and Lucy is not the winner. Therefore, Jasmine has the perfect strategy and will always win after all. □

Then without any hesitation, let us get started on our first game, the Liar Game!

2. The Liar Game

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

Let us play a piece-of-a-cake[2] "guessing" game.

Game. Suppose Lucy is given a range of numbers from 1 to n by our Jasmine, and she has to deduce the correct number x through a series of maximum q queries of the form: "Is x in S?" where S is a subset of $\{1, \ldots, n\}$. Jasmine is rather a sly person, so she can lie at most k times to the queries. The integral question is, "For what n, k, q can she deduce the correct number every time?"

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

Example 2.1. Lucy is given the 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 strategy that is guaranteed to* always *win 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 consider the worst-case scenario. That is, when Lucy queries a set S, x is always in the larger set between S and its complement set S'. Then, the best Lucy can do is to query a set S so that S' has the same size[3]. After one round, the remaining set has size $\lceil n/2 \rceil$. Thus, as we successively query q rounds, each time dividing the set into roughly two equal sizes, in the end, we have $\lceil n/2^q \rceil > 1$ [GKP94, pp. 71-2]. We can see that there is 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.

Suppose Jasmine picked $x = 3$ again.

Then Lucy can apply the strategy to always systematically query the first half of the set[4]: $\{1, 2\}$. 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", revealing the answer herself. Thus, Lucy wins.

We managed to find an explicit strategy deterministically for $k = 0$ and $n \leq 2^q$, 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.

---

[2] more like a piece of *difficult* cake :)

[3] Binary search pops up again!

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

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!

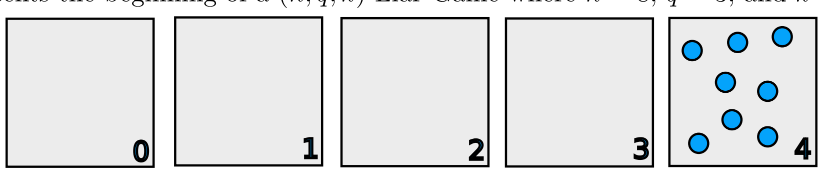
Game (The (n, q, k)-Liar Game)**.** Let us reframe the game into a two-sided game where Jasmine is also playing it to turn this into a sequential perfect information game. Jasmine will not pick an x in advance but rather use the information given by Lucy (through queries) to always answer consistently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

Game (The (n, q, k)-Chip-Liar Game)**.** Imagine a board of positions: $0, 1, \ldots, k$, 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 the beginning of a (n, q, k)-Liar Game where $n = 8$, $q = 3$, and $k = 4$.



Theorem 2.5. *The (n, q, k)-Chip-Liar Game is equivalent to the (n, q, k)-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 this is the same as Jasmine answering "No" to "Is x in S'?", thus they have to move S' to the left.

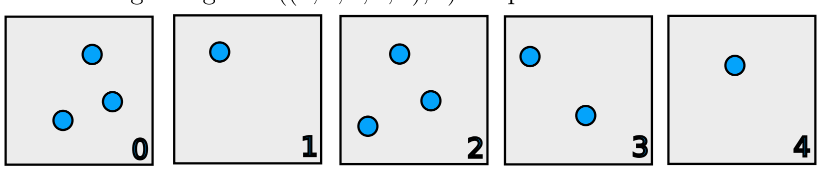
A chip i that has position j means that the question "Is i the number" has (potentially) received $k - j$ lies. Continuing the previous sentence, it is important to note that Jasmine can lie by saying either "Yes" or "No". Suppose a chip c is in the query set S, then when Jasmine lies and says "No", the chip c gets moved to the left. On the other hand, if the chip is in the complement set S', it can also get moved to the left when Jasmine lies and says "Yes". By enumerating two cases of the chip c, we can see that moving the chip c to the left by one exactly corresponds to Jasmine telling Lucy (possibly falsely!) that this chip is not the answer we're looking for.

The winning condition is also equivalent between the two games. A chip c not being on the board anymore means that Lucy does not have to consider the chip anymore. Thus, when the game has less than or equal to one chip, this means Lucy has successfully deduced the correct answer. When the game has more than one chip, this equates to Jasmine winning. $\square$

Before we figure out 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 $((x_0, \ldots, x_k), q)$-Chip-Liar Game where initially, the position i has x_i chips and $x_0 + x_1 + \cdots + x_k = n$. We then realize that when $x_k = n$ and other $x_i = 0$, the $((x_0, \ldots, x_k), q)$-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 the picture below represents the beginning of 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 is some uncertainty involved. How about Jasmine? How does she make sure that each "Yes" or "No" she answers to Lucy leads her to the situation that there is more than one chip at the end of the game?

Just as a student who forgets to study for her exam and circles the multiple choice randomly, perhaps, as a suggestion, we can model a "randomized" strategy for Jasmine and then show that there exists a "deterministic" strategy that works for her. By doing this, we rely on randomness to deduce a deterministic answer, thus, the essence of the probabilistic method.

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

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 combinations of flipping a coin q times and getting 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: $i = 0, 1, 2, 3, 4, \dots, j$:

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

Then we can introduce the theorem to answer our question.

Theorem 2.8. *If*

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

then Jasmine has a winning strategy for the $((x_0, \dots, x_k), q)$*-Chip-Liar Game.*

Proof. For Lucy, we let her play whatever strategy she has. First, focus 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 heads, and she moves S to the left if the coin is tails. For each chip c, let I_c be the boolean random variable for c remaining on board at the end of the game. 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 variable 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 prove that Jasmine would win the game with positive probability.

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 $\mathbb{E}[I_c]$.

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

The reason we want to figure out the boolean expected value of a chip is because of the linearity property of expectation (Theorem 1.1):

$$\mathbb{E}[X] = \mathbb{E}\left[\sum I_c\right] = \sum \mathbb{E}[I_c] = \sum_{i=0}^{k} x_i B(q, i).$$

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

By the initial assumption of the theorem's statement, $\mathbb{E}[X] > 1$. The expected value 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 remaining on the board at the end of the game, has positive probability to be larger than 1, Jasmine has positive probability to win.

The condition for Corollary 1.6 that Jasmine has positive probability to win is now satisfied. We then apply the corollary to conclude that Jasmine will always win. □

Corollary 2.9. *If*

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

then Jasmine has a winning strategy for the (n,q,k)-Liar Game.

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

Then to continue the discussion of $E[X]$ from the previous proof:

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

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

$$\begin{aligned}
n &> \frac{2^q}{\sum_{i=0} k\binom{q}{i}} \\
n2^{-q} &> \frac{2^q}{\sum_{i=0} k\binom{q}{i}} 2^{-q} \\
n2^{-q} &> \frac{1}{\sum_{i=0}^{k} \binom{q}{i}} \\
n2^{-q} \sum_{i=0}^{k} \binom{q}{i} &> \frac{1}{\sum_{i=0}^{k} \binom{q}{i}} \sum_{i=0}^{k} \binom{q}{i} \\
n2^{-q} \sum_{i=0}^{k} \binom{q}{i} &> 1 \\
n \sum_{i=0}^{k} 2^{-q} \binom{q}{i} &> 1 \\
\mathbb{E}[X] &> 1.
\end{aligned}$$

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 Jasmine always wins.

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].

Strategy 2.10 (for Jasmine in the $((x_0, \dots, x_k), q)$-Chip-Liar Game)**.** Suppose it is Jasmine's turn, that she has l moves remaining, that Lucy has just given the query "Is x in the set S?", and that there are currently y_i chips at position i. Define the weight of this position as $W = \sum_i y_i B(l, i)$.

- Let W^y be the new weight if Jasmine says "Yes" to the query "Is x in S" and moves S' to the left.
- Let W^n be the new weight if Jasmine says "No" to the query "Is x in S" and moves S to the left.

Then Jasmine chooses the answer which gives the maximum weight.

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 from now on.

Theorem 2.11 (Explicit Strategy for Chip-Liar Game)**.** *Suppose as in Theorem 2.8 that $W = \sum_i x_i B(q, i) > 1$. If Jasmine follows Strategy 2.10, then she will win.*

Proof. We will iteratively show that by following this strategy, at each step the weight $W > 1$. The base case is by assumption. Suppose Jasmine is in the situation of the strategy, with l moves remaining and current weight $W > 1$.

As Jasmine can only say "Yes" or "No" for an answer, suppose she was 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 a coin and then deciding whether she should answer "Yes" or "No".

Jasmine's plan for this game will revolve 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.

In particular, no matter what, following our maximum weight strategy at this step ensures the weight stays above 1.

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

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

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

Game (The $(x_1, \dots, x_k)$-Tenure Game)**.** Lucy wants to promote her faculty members to tenure, but Jasmine, her enemy, is in the admission council. There are k pre-tenure levels: $\{1, \dots, k\}$, with level 1 being the highest pre-tenure level and level 0 being the tenure level. Let each faculty member be represented by a chip. As the pool of applications has different tenure levels, this promoting game to tenurity begins with x_i chips level for $1 \leq i \leq k$ and no chips on level 0. 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 promoted a faculty to tenure and won the game. If none of the chips reaches level 0, i.e., no faculty becomes tenured, 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 theorem and a proof for this remark!

Theorem 3.1. *The $(x_1, \ldots, x_k)$-Tenure Game ends after k rounds, i.e., after k rounds we end up with either at least one tenured faculty or with 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 round of $(x_1, \ldots, x_k)$-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 the 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 $\sum_{i=1}^{k} x_i 2^{-i} < 1$ then Jasmine has a winning strategy for the $(x_1, \ldots, x_k)$-Tenure Game.*

Proof. As usual, we want to utilize the probabilistic method. First, let Jasmine play randomly against an arbitrary strategy from Lucy. Every round of promotion, after Lucy queries a set S, Jasmine flips a coin to make a decision. If it goes up heads, Jasmine promotes S' and fires S. If it goes up tails, Jasmine promotes S and fires S'. For each chip c in level i, let I_c be the boolean random variable for c reaching level 0. If $I_c = 1$, c is eventually tenured and conversely, if $I_c = 0$, then c is eventually fired. 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 is

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

Let $X = \sum I_c$. Then by the linearity of expectation (Theorem 1.1),

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

The above result shows that our initial assumption is the same as assuming $\mathbb{E}[X] < 1$. Therefore, 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 wins with positive probability.

Reaching the end of this proof, we can always try to use Corollary 1.6 to achieve the same result as the proof of Theorem 2.8. But why not introduce some challenges? Let us not rely on Corollary 1.6!

We see that even when allowing Lucy free range to use whatever strategy she might have at hand, she cannot always win. But Zermelo's theorem (Theorem 1.5) guarantees us a winner with her perfect strategy that always wins. This person cannot be Lucy, therefore, it must be Jasmine! $\square$

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

Strategy 3.3 (for Jasmine in the $(x_1, \dots, x_k)$-Tenure Game)**.** Suppose it is Jasmine's turn, that Lucy has just queried the set S for promotion, and that there are currently y_i chips at position i. Define the weight of this position as $W = \sum_i y_i 2^{-i}$.

- Let W^y be the new weight if Jasmine promotes the set S (and so move S to the left).
- Let W^n be the new weight if Jasmine fires the set S (and so moves S' to the left).

Then Jasmine chooses the answer which gives the minimum weight.

The weight stands for the expected value of Y, where Y is the random variable denoting the total number of chips at the end if Jasmine decides to play randomly from now.

Theorem 3.4. *If $\sum x_i 2^{-i} < 1$, then if Jasmine follows Strategy 3.3 she will always win the game.*

Proof. With the same logic as the Liar Game, $W = \frac{1}{2}(W^y + W^n)$.

Assuming the weight (expected value) to be less than 1, a similar argument can be made[5] just like Theorem 2.11 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 are zero chips at the end of the game. This results in Jasmine winning the game as there is no faculty becoming tenured. $\square$

"Wait a minute!" - a group of Lucifers[6] exclaims.

"I want Lucy to win!" - they angrily argue. "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 is forced to introduce a lemma and a theorem that reveals Lucy's winning play.

Lemma 3.5. *If a set of chips has a weight of at least one, it may be split into two parts, each of weight at least $\frac{1}{2}$.*

---

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

[6]Fans of Lucy.

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

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

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 $n = 2$, instead of 1, since we have shown above that no level can have only one chip. If the two chips are at some higher level $i > 1$, then $W = 2 * \frac{1}{2^i} < 1$, which contradicts the theorem's assumption. Then the two chips must be on the first position $i = 1$ so that $W = 2 * \frac{1}{2} = 1$. We can split them into two individual sets on the same position $i = 1$, consisting of only 1 chip, then the weight of each chip is $W = 1 * \frac{1}{2} = \frac{1}{2}$.

Suppose 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 the size of n, but with the same weight as the set S, with the only difference being that at position i where there are at least two chips, we compress the two chips into one superchip at position $i - 1$. We show the concept mathematically as follows. Let W^* be the weight of the set S^*, and z_t be the number of chips at position t of S'. Due to the compression of the two chips at level i, we have

$$z_t = \begin{cases} y_{i-1} + 1 & t = i - 1 \\ y_i - 2 & t = i \\ y_t & \text{else} \end{cases}.$$

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 the case n (which refers to S^*, which again refers to S).

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 transformed S to S^* and split S^* into two sets T and $T' = S^* \backslash T$ 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 number of chips returns back to $n+1$, and we have our set S back. The composition can be algebraically shown by backtracking from the last equation back to the first equation above.

This concludes the induction, therefore, the proof. □

We now provide a winning strategy for Lucy in the same format:

Strategy 3.6 (for Lucy in the $(x_1, \ldots, x_k)$-Tenure Game)**.** Whenever it is Lucy's turn and the weight of the remaining set of chips is at least one, Lucy can use Lemma 3.5 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 $\frac{1}{2}$. She can then query either set T or T' to Jasmine.

Theorem 3.7. *If $W = \sum x_i 2^{-i} \geq 1$ then if Lucy follows Strategy 3.6 she will always win the $(x_1, \ldots, x_k)$-Tenure Game.*

Proof. The assumption of the initial weight makes it possible for Lucy to employ Strategy 3.6.

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

Without loss of generality, suppose Jasmine promotes T and fires T', then each set of chips y_i at position i[7] 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, where y_i is the number of chips in T at position i before the promotion and at position $i-1$ after the promotion. Then

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

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 one throughout the whole game. This allows Lucy to reuse her strategy into the next round.

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 implies the same chip being at level 0. Thus Lucy successfully promotes her faculty to tenure and wins the $(x_1, \ldots, x_k)$-Tenure Game. □

4. Implementations

Readers interested in implementing these ideas can visit `https://github.com/badumbatish/Twoples_Proj` to see my primitive implementations in Python's Jupyter Notebook.

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

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

References

[GKP94] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. 2nd. USA: Addison-Wesley Longman Publishing Co., Inc., 1994. ISBN: 0201558025.

[vNM07] John von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. anniversary. Princeton University Press, Princeton, NJ, 2007. xxxii+739. ISBN: 978-0-691-13061-3. URL: https://mathscinet.ams.org/mathscinet-getitem?mr=2316805 (visited on 12/19/2022).

[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