Team Pyramid Scheme
Bernard, Eddie, Lucy, Zhang Liu

Team Mission Statement

Team Name

Pyramid Scheme

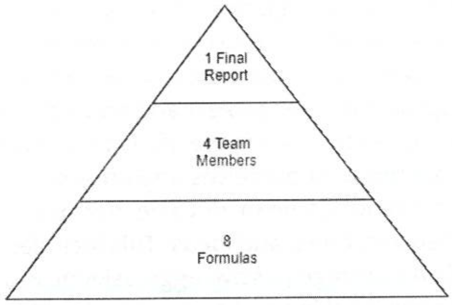


Figure 2: A Pyramid Scheme.

Preferred Contact

(preferred names in bold)
Bernard Boey: +65 9117 5827 (WhatsApp / Telegram), bernard@u.yale-nus.edu.sg
Eddie Lim: +65 9859 8323 (WhatsApp / Telegram), eddie.lim@u.yale-nus.edu.sg
Lucy Xinyu Zhu: +65 9448 3008 (WhatsApp / Telegram), xinyu.zhu@u.yale-nus.edu.sg
Zhang Liu: +65 8515 6883 (WhatsApp), zhangliu@u.yale-nus.edu.sg

Meeting Schedule

29 Jan, Wednesday 8:30 - 10:30pm	Question 1
30 Jan, Thursday 9 - 10:30am	Questions 2 & 3
6 Feb, Thursday 9 - 10:30am	Question * and review all questions
9 Feb, Sunday 9am - 1pm	Review full report and activity log

Project Approach, Working Together & Resolving Disagreements

- Before we meet
  - Complete the work we agreed to do before the meeting
  - Communicate in group chat
- When we meet
  - Set goals for the meeting session
  - Enact our practices while solving problems in class - discuss and visualize
- Resolving Disagreements
  - Seek to understand - Each person should explain their point of view / solution. This is in the hope that constructive comments can be made on each person's point of view, in order to find common ground.

This is somewhat generic. They are good basic principles for teamwork, but it would have helped to reference the specific project at hand.

Team Pyramid Scheme
Bernard, Eddie, Lucy, Zhang Liu

Guidelines for Interaction

- We recognize that this class is an elective.
- Everyone who is here wants to be here.
- We agree to do our best to make this a brave space to speak openly and share ideas for approaching this project and solving problems.
- We agree to listen respectfully to one another in the spirit of learning and to suspend judgement, of ourselves and others.
- We will not demean, devalue, nor put down other people for their various experiences, interpretations, and ideas. This includes avoiding the use of micro-aggressions.
- We discourage passive aggressiveness.
- We value asking and answering "stupid questions."
- We share a collective responsibility in our final submission - reviewing each other's work is a must.
- We recognize that we will each try our best, so we will be patient when needed.
- We will be punctual to our meetings.

List of Criteria for Evaluation

- The team member contributed (less/more than) a fair share of ideas to the project.
- The team member contributed (less/more than) a fair share of organization to the project.
- The team member contributed (less/more than) a fair share of writing to the project.
- The team member participated (less/more effectively) during the problem-solving sessions.
- The team member contributed (less/more than) a fair share of feedback on the report.

Signatories

Bernard Boey	Eddie Lim	Lucy Zhu	Zhang Liu

21 Jan 2020
Date

Team Pyramid Scheme
Bernard, Eddie, Lucy, Zhang Liu

Activity Log for Project 1: A Combinatorial Melange

Team Pyramid Scheme
Bernard, Eddie, Lucy, Zhang Liu

Wednesday 29 Jan, 8:30pm - 12:00am

<table>
<tr><td>Members</td><td>All members</td></tr>
<tr><td>Highlights</td><td>
1. Listed the cases for n = 1, 2, 3, 4.
2. Decided to reduce the problem to the first pattern strings with n 0s and n 1s.
3. Discovered the recurrence relation; hypothesized and verified for n = 5 and above using an Excel sheet
4. Created a Python program to check for the number of legal strings for n <= 6
</td></tr>
<tr><td>Details (process, including approaches, examples productive failures)

Conjecture!</td><td>
<u>Listing Cases</u> How did you do this? Where is the evidence?
We first listed all the cases for the 8 patterns for n = 1, 2, 3, 4 and found that the corresponding cases are all 1, 2, 5, 12. With this in mind, we decided to focus on just one pattern, the first one investigating strings of n 0s and n 1s such that no initial segments have more 1s than 0s. We hypothesized that the remaining patterns were related to the first pattern by bijection, which we hoped to prove after we figured out the formula for the first pattern.

<u>Attempting to identify a formula</u>
Observing the strings of 0s and 1s that we listed, we noticed that the first digit is always 0 and the last digit is always 1 for all values of n. With this in mind, we set about finding a formula for the pattern using the following approaches.

1. Slotting 1s into a row of 0s (and similarly, slotting 0s into a row of 1s). Then each bucket in between the 1s can hold a certain number of zeros.
For example, 01001101 -> 0 (1) 0 (0) 0 (2) 0 (1) where the number in the parentheses is how many 1s are between the 0s. The sum of the numbers in buckets cannot exceed the number of 0s which come before the bucket.
2. Prevent the “scale” from toppling over. We tried to set 1 as a weight and 0 as no weight. We weren’t able to come up with a rule which establishes a bijection to the 0s and 1s.
3. Reduce the string to the 2(n-1) middle digits. Since the first digit is always 0 and the last digit is always 1 for values of n, the string is uniquely determined by the middle 2(n-1) digits.
For example, 001101 -> 0110
4. Wrote down the cases which did not work. Since we couldn’t intuitively find a formula for the “legal” cases, we tried to take the approach of listing 2^n cases of binary strings, then finding a formula for the “illegal” cases.
</td></tr>
</table>

However, none of these brute methods gave us a formula for a_n with n. They were all productive failures which provided hints for a recurrence relation (formula for a_n with a_{n-1}).

<u>Identifying Recurrence Relation</u>

1. As noted above, the first digit is always 0 and the last digit is always 1 for all values of n, for every legal string for n-1, there was at least one legal string for n for appending '0' to the front and '1' to the end.

   Example: The cases for n=2 are 0011 0101; by appending '0' and '1' to the front and end respectively, the cases corresponding to 0011 are 010011 and 001101 and the case corresponding to 0101 is 010101.

2. By observing if the 'scale topples over' (an above approach from attempting to identify a formula), we also noticed a relationship where within the string of n '0's and '1's, between the first and last digit (which are '0' and '1' respectively), any consecutive '01's can be flipped without resulting in a illegal pattern.

   Example: From 0101, we obtain 001011 from (1).
   Using 001011, we can flip the first '01' pair we encounter to obtain **010**011; and the second '01' pair we encounter to obtain 00**1101**1; and both the '01' pairs to obtain 0**10101.**

   We can also obtain this result from 000111, which we obtain from 0011 from (1). By flipping the '01' pair we encounter, we obtain 00**10**11, which we can proceed to further flip '01' pairs we in the string.

3. Together from (1) and (2), the n=3 cases of 0011 0101 gives us the n=3 cases of 000111, 001011, 001101, 010101, 010011.

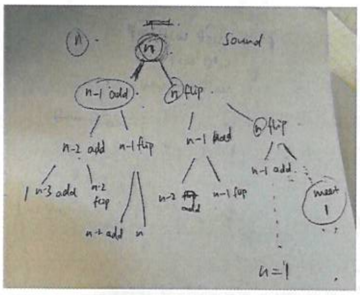
4. From here, we expanded the n=4 case and further realised that the nature of the relationships we have found in (1) and (2) allows us to categorise the strings by the number of '0's in the front of the string, which then expands recursively using the relationships.

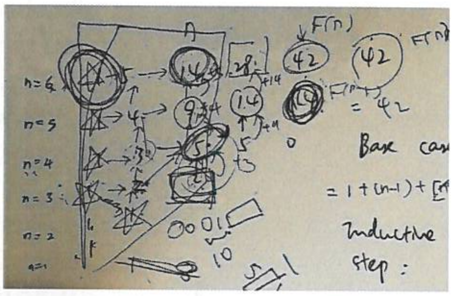
   Example: For n=3, we can categorise the strings into
   3 '0's at the front of the string: 000111
   2 '0s' at the front of the string: 001011; 001101
   1 '0's at the front of the string: 010101; 010011

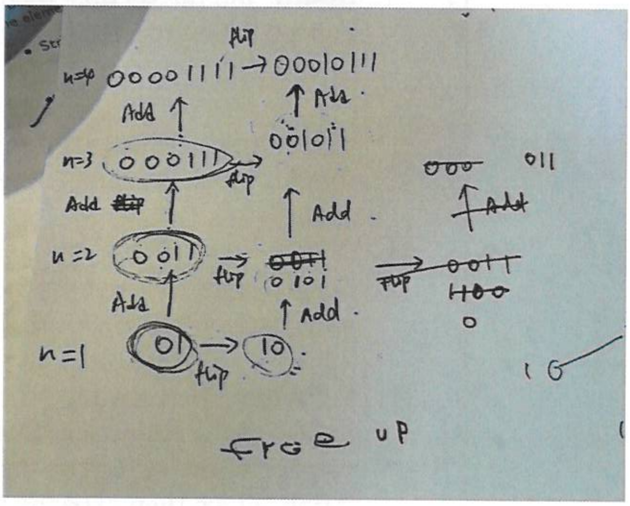
   For the n=4 case then, we expand on the results from the n=3 case to obtain:
   4 '0's at the front of the string: 00001111 (by appending)

3 ‘0’s at the front of the string: 00010111 (by appending), 00011011(by appending), 00011101 (by flipping)
2 ‘0s’ at the front of the string: 00101011 (by appending); 00100111 (by appending); 00110011 (by flipping); 00110101 (by flipping); 00101011 (by flipping)
1 ‘0’s at the front of the string: 01001011 (by flipping); 01010011 (by flipping); 01010101 (by flipping); 01000111 (by flipping); 01001101 (by flipping).

This gives us:
n=1 → 1
n=2 → 2
n=3 → 1 2 2 = 5
n=4 → 1 3 5 5 = 14







- We tried to verify this by inserting the logic into Excel to find the n=4, n=5. The formula used was to input a row of 1, 0, 0, ... as the bottom row. Enumerating rows and columns from left to right, from down to up as i and j, the [i, j] cell is the sum of cells [i, j-1] and [i-1, j].

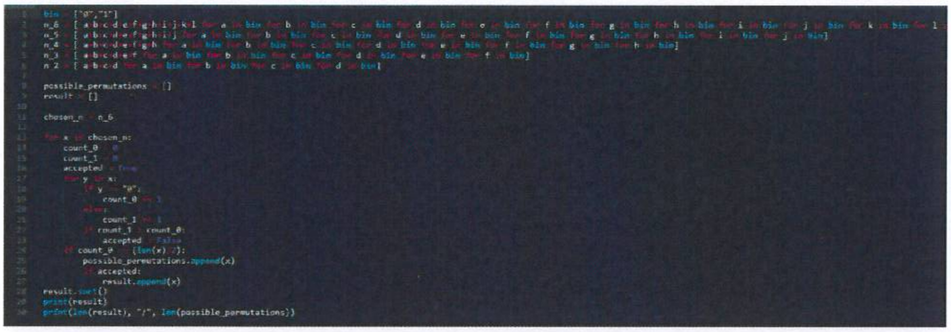
Team Pyramid Scheme
Bernard, Eddie, Lucy, Zhang Liu

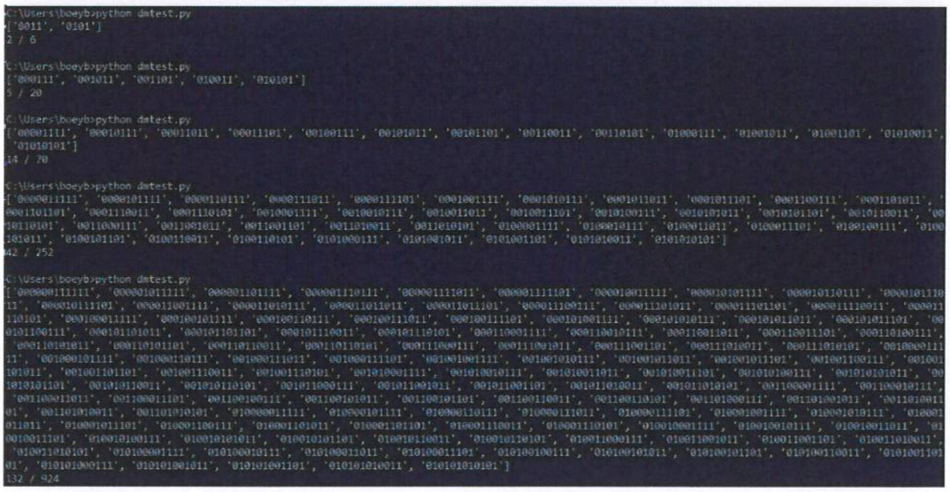
1	8	35	110	275	572	1001	1430	1430
1	7	27	75	165	297	429	429	0
1	6	20	48	90	132	132	0	0
1	5	14	28	42	42	0	0	0
1	4	9	14	14	0	0	0	0
1	3	5	5	0	0	0	0	0
1	2	2	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0

- Not only were we able to find the results for larger values of n, but we also noticed that the darker colored cell is the sum of the lighter colored cells. This deepened our understanding of the recursion. ✓

Automatic Enumeration of Legal Strings

We wrote a python program to identify all legal strings for n <=6 and the total number, in order to confirm that the numbers we described in the excel sheet were indeed correct





Attempting to Arrive at Formula for Recurrence Relation

- We briefly tried to arrive at a formula for the recurrence relation but were unable to.

what held you up?

Team Pyramid Scheme
Bernard, Eddie, Lucy, Zhang Liu

Connections to course material	The triangle of numbers which were enumerated using recurrence relations look similar to the pascal triangles in Chapter 4: No Matter How You Slice It.
Questions	1. What is the general formula for the recurrence relation identified? 2. What are bijections for the strings of 01s so that we can enumerate them in a non-recursive way? 3. Are the other examples indeed related to the first, or is it coincidental that they share the same results for n= 1, 2, 3, 4? 4. Is the recurrence relation related to the Fibonacci sequence or Pascal's triangle? How might the general formula for Fibonacci guide us to find a general formula for this recurrence?

Thursday, Jan 30, 9:00am - 10:30am

<table>
<tr><td>Members</td><td>All members</td></tr>
<tr><td>Highlights</td><td>
1. Discovered closed formula
2. Used bijection to justify closed formula
</td></tr>
<tr><td>Details (process, including approaches, examples productive failures)</td><td>
<u>Interpretation of charge and discharge</u>
To understand the meaning of “no more 1s than 0s,” we came up with the interpretation of 0s meaning discharge and 1s meaning charge. Then any sequence which accumulates 2 or more units of charge is illegal.
Example: A legal sequence 0101 would mean d-c-d-c, corresponding to the charge sequence of 0101. On the other hand, an illegal sequence of 0110 represents d-c-c-d and corresponds to the charge sequence of 0121. ✓

<u>Attempting to Arrive at Formula for Recurrence Relation (Productive Failure)</u>

We continued to attempt to find a formula for the recurrence relation. One thing we tried was to sum up the components on either side of the relation for n = 1, 2, 3, ... But we realised that it was too complex and we did not have the required knowledge to find an explicit formula.
How would you know what is required or not?

<u>Discovery of closed formula</u>

Looking at the numbers generated by the python program, we realised that the total number of combinations (legal + illegal) is a multiple of the legal combinations.

In particular,

<table>
<tr><th>n</th><th>Legal</th><th>Illegal + Legal</th></tr>
<tr><td>1</td><td>1</td><td>2 = 1 x 2</td></tr>
<tr><td>2</td><td>2</td><td>6 = 2 x 3</td></tr>
<tr><td>3</td><td>5</td><td>20 = 5 x 4</td></tr>
<tr><td>4</td><td>14</td><td>70 = 14 x 5</td></tr>
<tr><td>5</td><td>42</td><td>252 = 42 x 6</td></tr>
<tr><td>6</td><td>132</td><td>924 = 132 x 7</td></tr>
</table>
This allowed us to hypothesise a closed formula. Finding the total number of illegal + legal moves is trivial (2n choose n). Then, combining with the above knowledge, we hypothesised that the general formula is (2n choose n)/(n + 1). All that was left to do was to justify it.
</td></tr>
</table>

why is this trivial?

	<u>Alternative: Bijection</u> Since our recurrence relation method did not work, and further armed with our hypothesised formula, we decided to explore bijection. In what sense? We tried relating legal sequences to illegal sequences. Perhaps illegal sequences could be <u>flipped</u> from the position that the illegal move occurs in order to arrive at legal sequences. Then, illegal sequences would be some sort of complement to legal sequences. However, this could not be done easily. In particular, it was noted that new illegal moves could pop up due to the flipping. How would you know? In addition, we realised that trying to convert legal to illegal sequences or illegal to legal sequences <u>might be futile</u> since there is no one-to-one mapping, as seen in the examples of n = 2 to 6. Eventually, we tried flipping elements after an illegal move occurs. This results in a sequence with n + 1 1's and n - 1 0's with an illegal move. This approach seemed hopeful, but we were not sure how to show that this was a bijection. Then, we realised that with n + 1 1's and n - 1 0's, an illegal move was guaranteed since there would be more 1's than 0's. Hence this means that illegal n 1's n 0's sequences don't just map to illegal n+1 1's n-1 0's sequences, they map to **all** of the n+1 1's n-1 0's sequences. This was a huge breakthrough and we managed to find the formula pretty easily. The total possible sequences (both legal and illegal) are (2n choose n). The total sequences with n+1 1's and n-1 0's are (2n choose (n + 1)). This is also the number of illegal sequences. Hence, the solution is (2n choose n) - (2n choose (n + 1)) = (2n choose n)/(n + 1). ✓
Connections to course material	Calculation of the closed formula required understanding of Definition 3.15 from Chapter 3: There are a lot of them.
Questions	1. What are the legal and illegal transformations for the 7 other problems? ✓ 2. How might geometric shapes, such as in examples 3, 4, 8 be uniquely represented by the binary sequence, so that we can use our enumeration method for example 1? ✓

Thursday, Feb 6, 9:00am - 10:30am

Members	All members
Highlights	Discovered formulas and justification for problem 2 and 3
Details (process, including approaches, examples productive failures)	<u>Applying Bijection to Other Examples</u> • <u>Ways of multiplying an ordered list of n+1 numbers (Productive Failure)</u> ◦ We initially constructed the bijection using the opening and closing parentheses, where the opening parenthesis represented 0 while the closing parenthesis represented 1. ◦ We realised, however, that this could lead us to the same string in the cases of *((ab)c)d)* and *(a((bc)d)* for instance. ◦ Since we are concerned with the possible ways of multiplication, we realised that the numbers, rather than closing parentheses should be represented. This is so as numbers after an opening parenthesis and before the next opening parenthesis would already indicate which numbers the opening parenthesis apply to, without the need for the closing parentheses to be represented. ◦ Therefore, we could construct the bijection instead by representing the opening parenthesis as 0 and the number as 1. • <u>Ways in which 2n people seated around a circular table can all simultaneously shake hands with another person without crossing arms</u> ◦ The initial approach was to label each person around the table as 0 and 1 (010101 in a clockwise direction for n = 3). The idea was to constrain the pairs such that only 0-1 pairings can be made so that each '0' is accounted for by a '1' (as in the first example of the strings of n '0's and '1's). ◦ We tried finding a combinatorial solution for this, but it was too difficult to find. Why? ◦ We also realised that it does not sufficiently capture the location of each pairing to form a relationship. E.g. in 000111, to account for the 'charge' and 'discharge' (mentioned above) the positions 3 and 4 are a pair, 2 and 5 are a pair, and 1 and 6 are a pair. ◦ By labelling each point uniquely, e.g. by using A-F labels for n=3, we can then capture the relationship to construct the bijection.
Connections to course material	Constructing the bijections required understanding of Definitions 3.8 and 3.9, which states the criterion of injectivity and surjectivity.
Questions	1. What about the 5 other problems?

Thursday, Feb 6, 1:00pm - 3:00pm

Members	All members
Highlights	Discovered solutions to all other problems

Details (process, including approaches, examples productive failures)	At this point, we conjectured that the formula for all the problems were the same. Hence, we focused on finding bijections to each other. Yay!
	
	• Full rooted binary trees with n internal nodes
	  ○ First, we tried to label each node (except the root) with a 0 and 1, depending on whether they were on the left or right. However, we ran into problems with putting the sequence together. We were not sure what order to select the elements. We tried putting it together from left to right, top to bottom, but it did not work.
	  ○ Then, we realised that the solution was much simpler, as a binary tree could be represented with a string containing parentheses. Hence, we knew that we could construct a bijection between this and problem 2.
	
	• Paths along edges of an n x n grid which do not pass below the diagonal
	  ○ The solution for this was pretty obvious, once we realised that the moving upward could be represented by '0' and moving rightward could be represented by '1'.
	
	• Ways to tile a stairstep shape of height n with n non-overlapping rectangles
	  ○ This problem initially seemed difficult to solve because of the many different ways that the rectangles could be configured
	  ○ However, after we realised that binary trees could be linked to this object, the solution seemed simple.
	  ○ At first, we tried placing the nodes within the rectangles. However, that was unsystematic and was not a convincing solution. Eventually, we realised that the edges of the rectangles could represent the lines and the corners could represent nodes.
	
	• The ways in which numbers 1, 2, ..., 2n can be arranged in a 2 x n rectangular grid so that every row and column is increasing.
	  ○ At first, we tried swapping the elements if an illegal move was encountered. However, we realised that there may still be multiple illegal moves in the row.
	  ○ We realised that since there are 2 rows, we could assign each row a different number. With the first row as '0', and second row as '1', a bijection with the first problem could be constructed.
	  ○ However, later on, we realised that some illegal permutations had legal representations

<table>
<tr>
<td></td>
<td>

For example the following has 00001111 as the representation, which is a legal sequence although the configuration below is illegal
<table>
<tr><td>1</td><td>2</td><td>4</td><td>3</td></tr>
<tr><td>5</td><td>6</td><td>7</td><td>8</td></tr>
</table>

Fortunately, we realised that this was not actually a problem. This is because the legal sequences actually correspond to a different legal configuration

<table>
<tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>5</td><td>6</td><td>7</td><td>8</td></tr>
</table>

Furthermore, if we restrict the numbers to be strictly increasing for each row, we can remove the problem of legal sequences matching illegal configurations.

<u>Ways of cutting a convex polygon with n + 2 sides into triangles without adding new vertices</u>

At first, we thought that for any convex polygon with n + 2 sides, forming all triangles would result in 2 triangles with 2 edges and n - 2 triangles with 1 edge. Then it would be merely a matter of figuring out the locations of the 2 triangles and permuting them. Hence, a sequence was constructed (2 2 1), and a bijection was hypothesised. 2 2 1 would mean that going clockwise around the polygon, the first 2 sides form a triangle, the next 2 sides form a triangle, and the last side forms a triangle with internal sides. Thus, permuting this sequence would provide the answer.
However, we later realised that this was not the case. For a convex polygon of 6 sides, other combinations could be created. For example, 4 triangles could be formed where 3 triangles are made up of 2 external sides, while 1 triangle is made up of 3 internal sides.
Later on, we realised that all triangles are either made up of external sides or new internal sides that are derived from connecting external sides. This recursive structure could be represented by multiplying together variables which represented each side. Then, a bijection could be constructed between problem 2 and the ways of cutting a convex polygon.

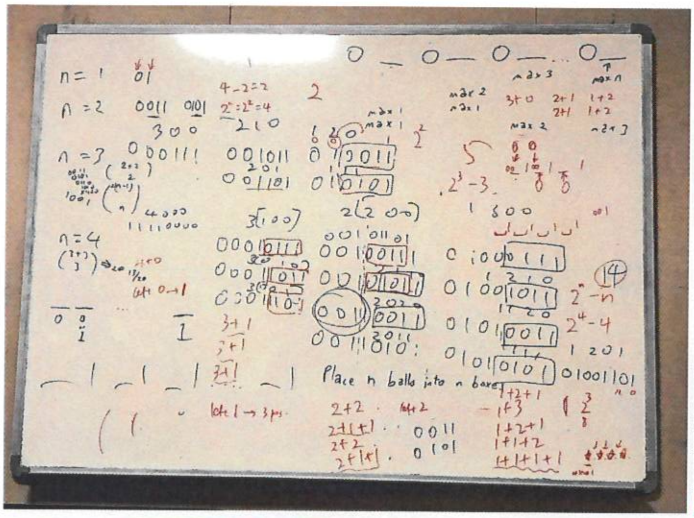
</td>
</tr>
<tr>
<td>Connections to course material</td>
<td>Constructing the bijections required understanding of Definitions 3.8 and 3.9, which states the criterion of injectivity and surjectivity.</td>
</tr>
<tr>
<td>Questions</td>
<td>1. How can we write formal proofs and prove bijections without relying on “n = 3” examples?</td>
</tr>
</table>

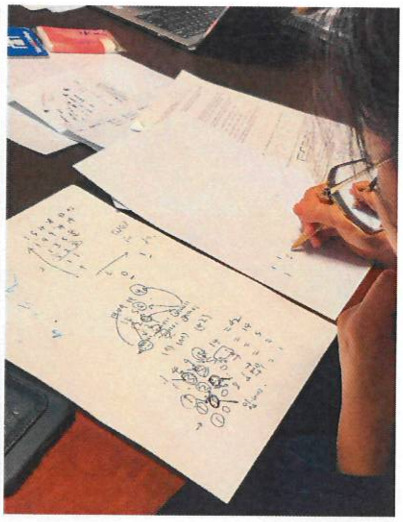
Sunday, Feb 9, 9:00am - 4:00pm

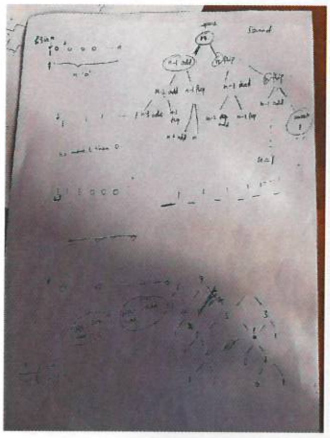
Members	All members
Highlights	Finished up report and cleaned up activity log
Details	• Provided formal proofs for each bijection • Discovered and polished the common properties among the formulas. • Inserted and formatted our solutions and conclusions in LaTeX format • Inserted images, screenshots and furnished more details for the activity log
Connections to course material	Constructing the bijections required understanding of Definitions 3.8 and 3.9, which states the criterion of injectivity and surjectivity.
Questions	1. Are there any exceptions to our discoveries? 2. Have we identified all the common properties of the objects specified? 3. How is what we identified useful for other combinatorial questions? What transferable skills have we developed along the way?

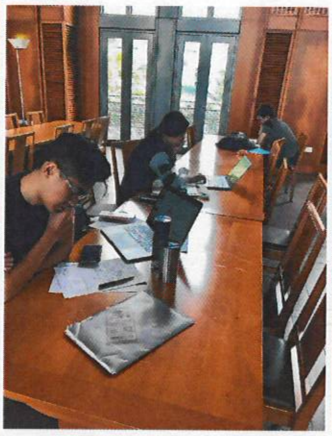
What was this process like?

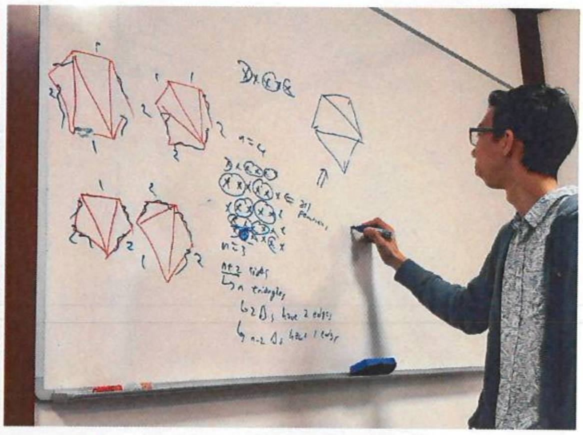
Selected photos of meetings and milestone drafts

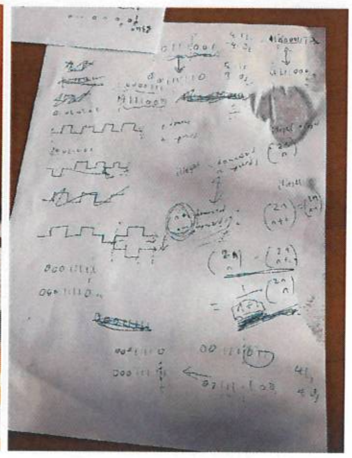












PROJECT 1: A COMBINATORIAL MELANGE

How many ways can we form pyramids?
A look into the recursive~~ly~~, binary and unidirectionality of combinatorial objects.

What does this mean?

Lucy Xinyu Zhu
Eddie Lim
Bernard Boey
Zhang Liu

February 9, 2020

Contents

Introduction

This report presents the team's discovery of how eight seemingly unrelated combinatorial objects share the same underlying structure. Our main approach to finding the formulas is to use the 01-string object as a starting point and "re-interpret" the other objects in relation to the 01-string by establishing bijections. In search of the fundamental properties, we start by noticing the initial pattern and establish connections by comparing the objects. We demonstrate that there are three major properties that all of the eight objects share: recursive structure, equivalence to binary strings, and constraints on directions.

What are these?

In this report, we first present how we reached our formulas for enumerating each of the eight combinatorial objects. Then, we explain how the formulas are related to each other. Thereafter, by some of the common properties of the objects, we justify the relationship we found between the formulas. Finally, we provide more examples of combinatorial objects that belong to this family, revealing an elaborate scheme. ;-)

The reader would have to reference the project brief to make sense of this. If that's the case, you should use the same language/notation as the brief.

Question 1

(1) The formula for the number of strings of n 0's and n 1's such that no initial segments have more 1's than 0's is $\frac{1}{n+1}\binom{2n}{n}$.

To derive this formula, we construct a bijection from the set of all strings of n 0's and n 1's which do not fulfill the property above (which we will term as illegal strings in this report) onto that of all strings of $(n-1)$ 0's and $(n+1)$ 1's.

To construct the bijection, let B be an illegal string of n 0's and n 1's and C be a string of $(n-1)$ 0's and $(n+1)$ 1's. We define $f(B)$ using the following process:

(a) In ~~a~~ B ~~string~~, find the ~~ith~~ first position where ~~the~~ an "illegal move" occurs. This is the first position where more 1's appear than 0's considering the initial segment from the start of the string to the ith position. (E.g. In the illegal string 01**1**001, the digit in bold is the first illegal move, and hence the ith position). $i = 3$?

(b) Flip all digits after the ith position. (E.g. The string will now be 01**1**110) This is exactly a C string.

Let $f(B)$ be the resultant string, which contains $(n-1)$ 0's and $(n+1)$ 1's. Given any $f(B)$ string with $(n-1)$ 0's and $(n+1)$ 1's, we can find the unique illegal string B with n 0's and n 1's. This is because in a C string with 2 more 1's than 0's, the string will definitely contain an illegal move. Flipping all digits after the illegal move in a C string will result in a B string with n 0's and n 1's which contain at least one illegal move. This is exactly B. Hence, this shows a unique mapping of B and C.

Why?

?

What does this mean?

What if there are more?

Why is this mapping injective?

The mapping of strings with n 0's and n 1's to strings with $(n-1)$ 0's and $(n+1)$ 1's:

n 0's, n 1's	$(n-1)$ 0's, $(n+1)$ 1's
000111	legal
001011	legal
001101	legal
001110	001111
010011	legal
010101	legal
010110	010111
011001	011110
011010	011101
011100	011011
100011	111100
100101	111010
100110	111001
101001	110110
101010	110101
101100	110011
110001	101110
110010	101101
110100	101011
111000	100111

Why? is the "clearly" needed here?

The total number of strings with n 0's and n 1's is clearly $\binom{2n}{n}$, as we have $2n$ digits and we can construct any string by choosing n digits to be 1 (or choosing n digits to be 0). Hence this is equivalent to choosing n-element subsets from $[2n]$.

The total number of strings with $(n-1)$ 0's and $(n+1)$ 1's is clearly $\binom{2n}{n+1}$, as we have $2n$ digits and we can construct any string by choosing $n+1$ digits to be 1 (or choosing $n-1$ digits to be 0). Hence this is equivalent to choosing $(n+1)$-element subsets from $[2n]$. This is also the total number of illegal strings with n 0's and n 1's, as shown earlier. Hence the total number of strings ~~with~~ n 0's and n 1's that fulfills the above property is $\binom{2n}{n} - \binom{2n}{n+1} = \frac{1}{n+1}\binom{2n}{n}$

why?

(2) For $n+1$ numbers, there are corresponding n pairs of parentheses that can be placed to multiply the ordered list of $n+1$ numbers.

The ways of multiplying an ordered list of $n+1$ numbers together is uniquely determined by the position of the parentheses and the numbers being multiplied - each pair of parentheses changes the order to which the numbers are multiplied. As such, we examine the pairs of parentheses and the numbers that ~~the~~ are contained within each parenthesis. To do so, we only need to observe the location of the opening parentheses and the numbers. That is, if the i-th position of the string is a opening parenthesis, then let the i-th position represent 0 while if the i-th position of the string is a number (e.g. a, b, c), then let the i-th position represent 1. Due to the

?

nature of parentheses grammar, the opening parenthesis must appear in the expression before the numbers contained in the respective parentheses. In other words, no more 1's than 0's are allowed in any initial segment. We can then construct a bijection from the set of all distributions of the opening parentheses and numbers onto that of the first example, strings of n 0's and 1's such that no initial segments have more 1's than 0's. Note that the last $(n+1)$th number (e.g. d in $n = 3$) is irrelevant and not counted because for a given set of parentheses and numbers up till n, the position of the $(n+1)$th number has already been determined (i.e. the multiplication symbols have already been positioned and one number is missing to complete the expression. There is no other way to position the last number).

$(((ab)c)d)$	$((ab)(cd))$	$((a(bc))d)$	$(a((bc)d))$	$(a(b(cd)))$
$(((xxx$	$((xx(x$	$((x(xx$	$(x((xx$	$(x(x(x$
000111	001101	001011	010011	010101

FIGURE 1

This is then clearly a bijection to the first example, as any string of n 0's and 1's can be obtained from exactly one distribution of parentheses this way. We can then obtain the formula for example 2 as $\frac{1}{n+1}\binom{2n}{n}$.

How do you know?

This needs justification.

(3) By labelling each person uniquely and reconstructing the diagram linearly (see Figure 2 below), we can construct a bijection from the set of all distributions of simultaneous handshakes with no arms crossing onto the strings of n 0's and n 1's such that no initial segments have more 1's than 0's, like in the first example. That is, if the ith person comes before their partner in the enumeration of all the people (i.e. on the left of the pair in the reconstructed diagram), let the i-th person be a 0, and if the i-th person is the second of the pair (i.e. on the right of the pair in the reconstructed diagram), let the ith person be a 1.

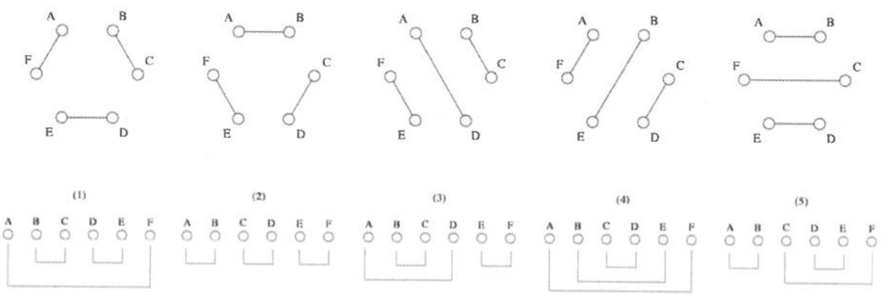


FIGURE 2

This is then clearly a bijection to the first example, as any string of n 0's and 1's can be obtained from exactly one distribution of the people

How do you know?

If this is so clear, you should be able to write (1) why it is a function between the two sets.
(2) why it is injective.
(3) why it is surjective.

sharing simultaneous handshakes without crossing arms this way. We can then obtain the formula for example 2 as $\frac{1}{n+1}\binom{2n}{n}$.

(4) The formula for the number of ways of cutting a convex polygon with n + 2 sides into triangles without adding new vertices is $\frac{1}{n+1}\binom{2n}{n}$.

How do you know?

Note that the number of triangles formed in a convex polygon without adding new vertices is always $n+2-2=n$. Furthermore, every single triangle has at least 1 side that ~~are~~ is formed by new lines formed by connecting existing vertices.

How do you know?

We construct a bijection from the set of ways of cutting a convex polygon with $n+2$ sides into that of all ways of multiplying an ordered list of $n+1$ numbers together.

To construct the bijection, let B be a method of cutting a convex polygon. We define $f(B)$ using the following process:

We label each side of the polygon (except the base) with a letter, and then we can describe a new line by multiplying the two letters together. For example, the line formed when connecting sides a and b is (ab). Notice that we form a triangle when we do this. On the other side, the line formed when connecting sides c and d is (cd). We keep connecting 2 sides until we reach the base of the polygon. In this example, with the 2 new lines, we can form a new triangle with the base of the polygon. This base can then be described as $((ab)(cd))$. This is the representation of all the cuts made on the convex polygon.

What if there are no triangles with two exterior sides?

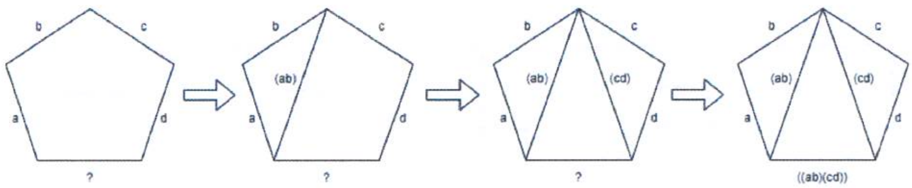


FIGURE 3

How do you know this process terminates at the base side

Note that each letter (side) appears exactly once because each outer side must form 1 and only 1 triangle and subsequent triangles can be formed using sides produced from the first triangle. Furthermore, each multiplication (or each left parentheses) represents a triangle in the polygon. Since there are $n+2-1=n+1$ letters in the representation, there are exactly $n+1-1=n$ multiplications (or left parentheses). This is exactly equal to the number of triangles in the polygon.

Hence, given a way of multiplying an ordered list of $n+1$ numbers together, we can find the unique way of cutting the convex polygon.

Why is it unique?

With the bijection set up, the number of ways of cutting a convex polygon with $n+2$ sides into triangles is equal to the number of ways of multiplying

This still needs to be shown.

an ordered list of $n+1$ numbers together which is $\frac{1}{n+1}\binom{2n}{n}$.

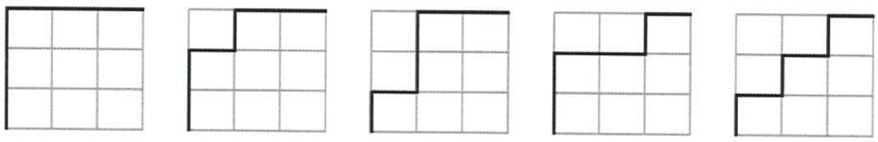
We can also think of the representation as traversing from the outer edges, through all the triangles and internal lines, to the base of the polygon. By traversing through the polygon, we would have covered all new sides and triangles formed, knowing the way that they were formed.

How do you know you end at the base?

(5) The formula for the number of paths along the edges of an n-by-n grid which do not pass below the diagonal is $\frac{1}{n+1}\binom{2n}{n}$.

Each path is formed by a series of moves going up for one grid and moves going right for one grid. The constraint is that the path cannot pass below the diagonal, which means the cumulative right moves should not exceed the cumulative up moves, at all times. Let the up move be "0" and the right move be "1." Then the problem becomes to write each path as a string of 0's and 1's. We just need to ensure that at any point of writing out the string, there are not more 1's (the right moves) than 0's (the up moves) before this point. With that, we can construct a bijection from the set of paths that do not pass below the diagonal to the set of 01-strings that have no initial segments containing more 1's than 0's. Therefore, the total number of such paths is the same as the number of strings of n 0's and n 1's such that no initial segments have more 1's than 0's, which is $\frac{1}{n+1}\binom{2n}{n}$.

Why is this a bijection?



000111 001011 010011 001101 010101

FIGURE 4

(6) Let the nodes which have nothing attached below them be called external nodes. There are $n+1$ external nodes for a binary tree with n internal nodes. Label the external nodes alphabetically from left to right, up to down and enclose those nodes which are connected to the same internal node within parentheses. We thus have established a bijection between this example and the second example, which is the ways of multiplying an ordered list of $n+1$ numbers together.

How do you know?

According to what arrangement?

Why is it a bijection?

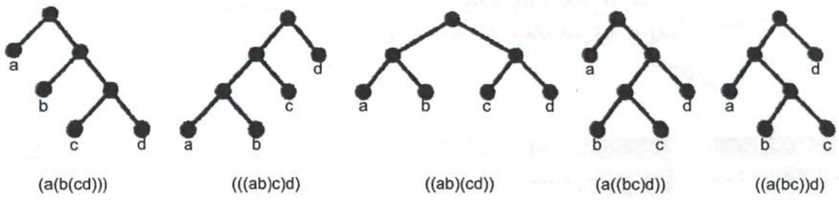


FIGURE 5

(7) Noticing that each number from 1 to $2n$ is either in the top row or bottom row, let the top row be 0 and bottom row be 1.

We first out the list of numbers from 1 to $2n$ in ascending order. This satisfies the condition that every row of the $2 \times n$ rectangular grid is increasing. Then we assign 0 to the numbers in the top row and 1 to the numbers in the bottom row. By stipulating that there cannot be an initial sequence with more 1s than 0s, this guarantees that we satisfy the condition of every column is increasing. Therefore, we have established a bijection between this example and the first example.

How do you know?

Why is it a bijection?

What rules?

?

Below are the bijections for $n = 3$. Note that we are only bijecting the cases which satisfy the rules onto example one. The bijections for legal cases in this example are unique. You may find our explanation for illegal cases in the Activity Log.

The report should not reference the log.

1	2	3
4	5	6

123456
000111

1	2	4
3	5	6

123456
001011

1	3	4
2	5	6

123456
001101

1	2	5
3	4	6

123456
010011

1	3	5
2	4	6

123456
010101

FIGURE 6

E (6)

(8) We can biject this example onto example ~~six~~ binary trees by stipulating that places where more than two corners of rectangles meet is an internal node. Places where a one rectangle's corner meets with another rectangle's side do not translate to a node.

This is difficult to parse.

?

?

It is an injection to example two because for each of the n rectangles, at least one of their corners meets the corner of another rectangle, creating n internal nodes. It is a surjection to example two because each internal node of a binary tree corresponds to two sides of a rectangle.

In Figure 7, if we tilt our head 45° to the left, we can see that each stairstep shape of height n corresponds to a unique binary tree with n internal nodes.

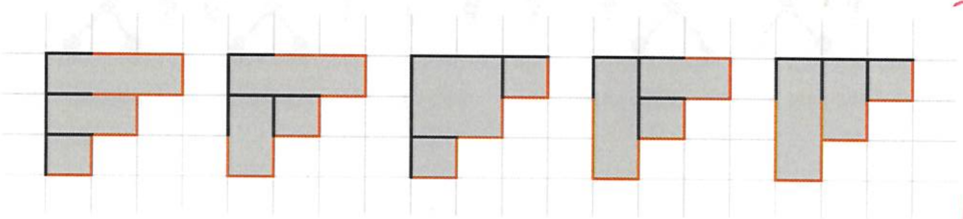


FIGURE 7

How does it work in general?

Question 2

The formulas found in Part 1 are exactly the same. We ~~can construct~~ constructed a bijection from enumerations for a problem to enumerations for another problem. More details on generalization of formulas and the underlying properties will be discussed in depth in the next section.

What is the purpose of this section?

?

Question 3

Having established the formulas, in this section we probe further and explore some properties of the underlying structure shared by all eight objects in Question 1. We will focus on three major properties, namely recursive structure, binary components, and constraints on direction.

(1) **Recursive structure.**
First, the structure for the n-th is the same as all previous $(n-1)$. Using the divide-and-conquer perspective we have learnt in the readings, the problem of finding the formula for n can be divided into sub-problems and taking the sum of all. We will illustrate this point further by using the 01-string (Object 1) and the square grid paths (Object 5) examples. We start with the pattern observed by listing the 01-strings as shown below.

What are they?

n = 4	00001111	00010111	00101101	01000111
		00011011	00100111	01001011
		00011101	00101011	01010011
			00110011	01010101
			00110101	01001101
n = 3	000111	001011	010011	
		001101	010101	
n = 2	0011	0101		
n = 1	01			

What is the general process for this?

By counting the number of strings in each cell and extending, we obtain the following pattern:

n = 6	1	6	20	48	90	132	132
n = 5	1	5	14	28	42	42	
n = 4	1	4	9	14	14		
n = 3	1	3	5	5			
n = 2	1	2	2				
n = 1	1	1					
n = 0	1						

We establish the connection by thinking about the n-by-n square grid in Object 5. The number of paths from $(1,1)$ to any point on the grid board is equal to the number of paths from $(1,1)$ to the point below it plus the number of paths from $(1,1)$ to the point on its left. For the square grid board, the number of paths from $(1,1)$ to (n,n) is thus the sum of all its sub-problems. As shown in Figure 8, this recursive structure is also found in enumerating the 01-string. All 01-string with n 0's and n 1's can be enumerated by repeatedly doing two kinds of transformation to the string:

1) adding 0 at the beginning of string with $n-1$ 0's and $n-1$ 1's and 1 at the end of it; and 2) flipping the symmetrical 0 and 1 from string with n 0's and n 1's.

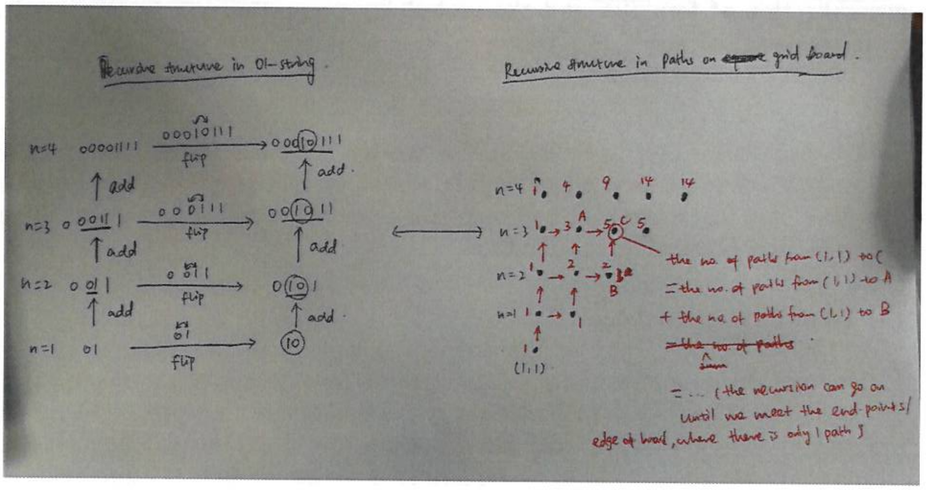


FIGURE 8. Recursive Structure.

So what is the recursive formula?

This should probably be in the activity log instead.

(2) **Equivalence to strings consisting of binary elements.**
Second, all eight objects, though appear in different forms, can be shown to be equivalent to strings consisting of two distinct basic building blocks. The key lies in choosing the right representation and interpretation. For instance, for the n-by-n square grid, we can think of the "up" moves as 0s and "right" moves as 1s (as explained for Object 5 in Question 1); for Object 2, we can think of the open parentheses as 0s and the numbers as 1s (as explained for Object 2 in Question 1).

What is the purpose of this section? You already established the bijections in an earlier section?

(3) **Constraints on direction.** Why direction?
Lastly, all the eight objects involve constraints on the direction. This property gives rise to the "illegal moves" that we have to exclude from the counting. There are many ways we can illustrate this point, but the best way to visualize is to contrast the n-by-n rectangular grid and the n-by-n triangular grid, as shown in Table 1 on the next page.

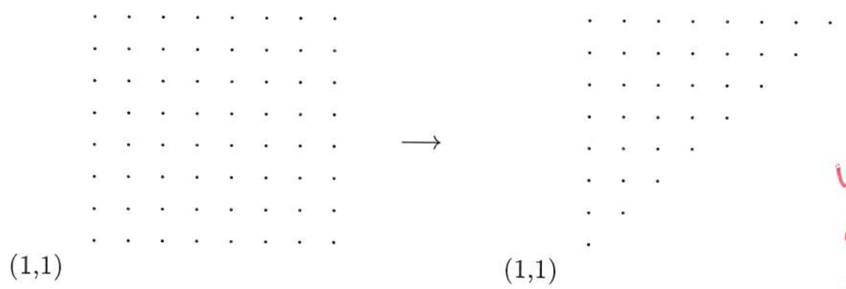


Table 1: Constraints on Square Grid.

What are the constraints on triangulations of polygons?

When we are **constrained** to only the upper left half of the board, our direction is also constrained in the sense that we now have to exclude the "illegal moves" that lead us outside the upper triangle. This is analogous to the "illegal moves" in the 01-string, as explained in our solution for Object 1 in Question 1.

*ANALOGOUS PROBLEMS

(1) Given an n-by$n$ triangular game grid, how many ways do we have if we want to move from the lower left corner to the upper right corner, provided that the only legal moves are moving up or moving right?

Why is this the same?

(2) How many possible distinct series consisting of only 1 and -1 such that any partial sum of the series is non-negative?

" "

(3) In Pingala's treaties (c. 3rd/2nd century BCE) on Sanskrit prosody, each line of the verse is consisted of strong syllables, Guru (denoted by **G**), and weak syllables, Laghu (denoted by **L**). Suppose in a particular ritual ceremony, each verse is not supposed to have more **G** syllables than **G** syllables at any time. For n **G** syllables and n **L** syllables, how many ways can we arrange the positions of **G** and **L** syllables?

" "

(4) How many ways are there to construct a path from the start point to the end point $2n$ units away? Here are the restrictions: The only legal moves are 1) moving upward then rightward by 1 unit, 2) moving downward then rightward by 1 unit. The path cannot go below the start/end points. There must be n upward then rightward moves and n downward then rightward moves. The possible paths are a bijection into the sequences of 0's and 1's from the first question. Enumerations for n = 3 can be found below, with an example of an illegal path.

" "

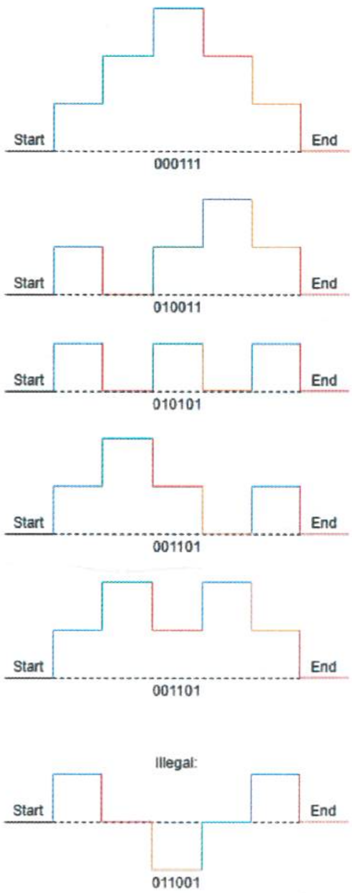


FIGURE 9

Interestingly, the possible paths heavily resemble pyramids. I guess we can say that we have discovered a *pyramid scheme.*

Conclusion

In this report, we have specified how eight seemingly unrelated combinatorial objects share the same underlying structure by investigating them one by one. Our gradational progress in the investigation along the way has allowed us to enquire deeply into the nature of these objects - from theorising its recurrence relation, to discovering the common properties shared by these objects which we have synthesised in the report.

What was this?

More importantly, by taking advantage of bijections with familiar objects to enumerate foreign objects, which is a powerful technique in Discrete Mathematics that we utilized multiple times in class, we were also able to expand the results across different combinatorial objects and scenarios.

Through our analogous problems, we have demonstrated multiple other objects that use the same formula, thereby expanding the list of objects known to demonstrate the common properties. In addition, we have independently confirmed the formulas for the enumeration problems, strengthening the work and proofs of previous mathematicians. It might also be possible that we have come up with alternative explanations and justifications for the formula of the enumeration problems.

What other mathematicians?

These fundamental properties could potentially be applied in objects that involve binary behaviors (e.g., alternating series) and recursive substructure (e.g., binary search). We suggest that further investigations should be conducted in light of similar recursive patterns and natural science occurrences of other sequences, such as the Fibonacci numbers.

As a team, we also felt that this project allowed us to exercise our abilities in proof, pattern recognition, creative problem-solving, and teamwork, while being a practical synthesis to what we have learnt in the classroom.

This is truly a *Pyramid Scheme*!

Bibliography

(1) Benjamin, Arthur T., and Jennifer J. Quinn. Proofs That Really Count: The Art of Combinatorial Proof. Washington, DC: Mathematical Association of America, 2003.

(2) Bona Miklos. A Walk through Combinatorics: an Introduction to Enumeration and Graph Theory. New Jersey: World Scientific, 2016.

Activity Log

The Activity Log, documenting our progress over the course of this project, is attached as a separate document.

YSC2213 DISCRETE MATHEMATICS

Project 1 Reflection and Peer Assessment

Zhang Liu
A0190879J

February 11, 2020

1. Reflection

- **General experience for Project 1.** I grealy enjoyed the process of working through the problems and eventually finding out a common strand that ties them all together. We also had a great team (with a great team name that turned out to be surpisingly somewhat prophetic).
- **What worked well.** I find that within our team, our skill sets complement each other pretty well. Bernard is good at coming up with concrete examples in a speedy and accurate manner; Lucy is very good at synthesizing and organizing information; Eddie is very good at spotting patterns; I find it natural to ask questions about connections like "why are they realted?" and "what is in common in their strucutre?" Therefore, we make use of our strengths while help each other out. When we have an idea but get stuck at one particular point, another person will come to the rescue with their set of skills.
- **What did not work well.**

  I think we could have met up more frequently and be more patien with the discovery process. I found that sometimes we wanted the results too eagerly and once we have obtained some results, we stopped thinking and reflecting on it. Although at the moment it seemed efficient not to chew on it, but often that would provide new insights that lead to better intuition for the next problem (and in future problems too.)
- **What I could have done better.**

  I could have communicate my ideas more clearly and more concretely. I realized that I had a tendency to get "inside my own head," approaching the problem in a way that is intuitive to me but is hard to communicate across. To be specific, in this project, I was overly obsessed with generalizing the problem using recursive relations and spent a lot of time outside meetings trying to come up with a rigorous generalization on my own. In the end, even though I understood the connections abstractly in my head, I did not do very well in putting those ideas in convincing explanations and connecting them with the other pieces of results we have come up with. As a result I had to discard some of those ideas.

  If I could be more clear and concrete in communicating those ideas to my teammates, our final product would have been more illuminating, for it would bring everything to a full circle with a common strand and a more rigorous/formal generalization. (But then again, we have to acknolwedge that we did the project within limited time and without external references. So perhaps I should be contented with what we have come up with.)
- **Personal goal for future objects.**

  I will make it a point to prune and communicate my intuitions, instead of relying on them and working out things abstractly in my own head. I will implement this by making a log of the list of the things I've thought about, in other words, a concise chronicle of thoughts. Hopefully by forcing myself to write out my ideas concisely and precisely, I will be able to explain more clearly to others.

Excellent!

2. Peer Assessment

List of Criteria for Evaluation:

- The team member contributed (less/more than) a fair share of ideas to the project.
  Bernard Boey: 2; Eddie Lim: 2; Lucy Zhu: 2.
- The team member contributed (less/more than) a fair share of organization to the project.
  Bernard Boey: 2; Eddie Lim: 2; Lucy Zhu: 2.
- The team member contributed (less/more than) a fair share of writing to the project.
  Bernard Boey: 2; Eddie Lim: 2; Lucy Zhu: 2.
- The team member participated (less/more effectively) during the problem-solving sessions.
  Bernard Boey: 2; Eddie Lim: 2; Lucy Zhu: 2.
- The team member contributed (less/more than) a fair share of feedback on the report.
  Bernard Boey: 2; Eddie Lim: 2; Lucy Zhu: 2.