

# Choices Made for Project 3

Mengxiang Jiang

December 17, 2025

## 1 Introduction

This was actually the third attempt at writing this project, since my first two attempts were on topics that just were too long to fit in the two page limitation. The first two topics were on the number of structural isomers of alkanes and the number of groups of order  $n$  respectively, in case you're curious. For this third attempt, I decided to go with a topic I was already familiar with from high school, which is block walking on Pascal's triangle. This topic is interesting because it combines combinatorics with a visual representation, and it allows for exploration of various patterns and properties within Pascal's triangle. One thing I disliked about many proofs of combinatorial identities is that they often rely on algebraic manipulations rather than intuitive reasoning. The proof technique used is also often induction, which I also find not very insightful. I will quote a passage from *The Art of Computer Programming Vol 1* by Donald Knuth here [2]:

Most textbooks would simply state those formulas, and prove them by induction. Induction is, of course, a perfectly valid procedure; but it does not give any insight into how on earth a person would ever have dreamed the formula up in the first place, except by some lucky guess.

Therefore, I wanted to explore combinatorial identities using block walking, which provides a more visual and intuitive approach to understanding these identities.

## 2 Some Choices Made

I tried to stick to many of the suggestions given by Paul Halmos in his *How to Write Mathematics* [1]. For example, I tried to keep the number of topics discussed to a minimum, and focused on just one topic and one theorem.

The audience I had in mind were high school or early undergraduate students who have some familiarity with combinatorics but may not have seen block walking before. As such I did not fully define binomial coefficients or Pascal's triangle since I assumed familiarity.

I also tried to be clear and concise in my explanations, but also anticipating potential difficulties by fully expanding an equation rather than just writing the final result in one tricky step.

And on the use and avoidance of repetition, I included a theorem at the end that is very similar to one proven, but I left the proof as an exercise to the reader. This helps emphasize the slight difference between the two theorems without having to repeat a proof.

One last thing that particularly resonated with me when reading Halmos was his anecdote about reading a label (1) in a book on page 89, but the display for label (1) was actually on page 90. He felt helpless and bewildered for five minutes until he finally figured it out. At the end, this trick left him feeling “stupid and cheated”, and Halmos vowed to never forgive the author. I tried to avoid such confusion in my writing by carefully labeling equations and figures, and referring to them in proper order to avoid disorienting the reader. I have also used the `hyperref` package to make cross-references clickable. This also creates proper links in the LaTeXML html web page. I hope this will save my reader’s time and will end on this note with a quote from Halmos:

If you work eight hours to save five minutes of the reader’s time, you have saved over 80 man-hours for each 1000 readers, and your name will be deservedly blessed down the corridors of many mathematics buildings.

### 3 Bibliography

### References

- [1] Paul R Halmos. How to write mathematics. *L’enseignement mathématique*, 16(2):123–152, 1970.
- [2] Donald E Knuth. *The Art of Computer Programming: Fundamental Algorithms, Volume 1*. Addison-Wesley Professional, 1997.