Chapter 14
Didactics of Combinatorics

ABSTRACT

The chapter focuses on teaching combinatorics, a challenging mathematical discipline to learn and teach. Its importance lies in developing logical reasoning, problem-solving skills, and applications in computer science, statistics, and probability. The suggested teaching approach emphasizes real-world problems, interdisciplinary connections, and technological tools. Active discovery, using games, simulations, and collaborative activities, is encouraged rather than mere memorization of formulas. Various examples illustrate how to make combinatorics more engaging. These include the "Pyramid Game," based on a dice roll and a triangular structure, and the use of the Fano plane to organize work shifts or optimize lottery strategies. Other problems include the "Tower of London," which enhances logical skills, and the carpooling problem, which explores different group arrangements. The goal is to integrate theory and practice, making combinatorics more accessible and stimulating for students.

DIDACTICS OF COMBINATORICS

A good teacher is a master of simplification and an enemy of simplism

Louis A. Berman

After examining the most relevant aspects of the topic, one can ask what the important elements are for those who have to teach this discipline to students, considering that, for Eizenberg and Zaslavsky (2004), "Combinatorics is considered one of the more difficult mathematical topics to teach and to learn". The study of combinatorics is important for several reasons: it helps to improve combinatorial reasoning (Kenney & Hirsch, 1991; Batanero et al., 1997; Hart & Martin, 2018; Medová et al., 2020; Simamora & Zunaiedy, 2021; Nopriana et al., 2023) and problem-solving skills, is helpful in many fields such as computer science, statistics, and probability theory, and forms a basis for more advanced mathematical concepts useful for achieving certain tasks (Semanišinová, 2021).

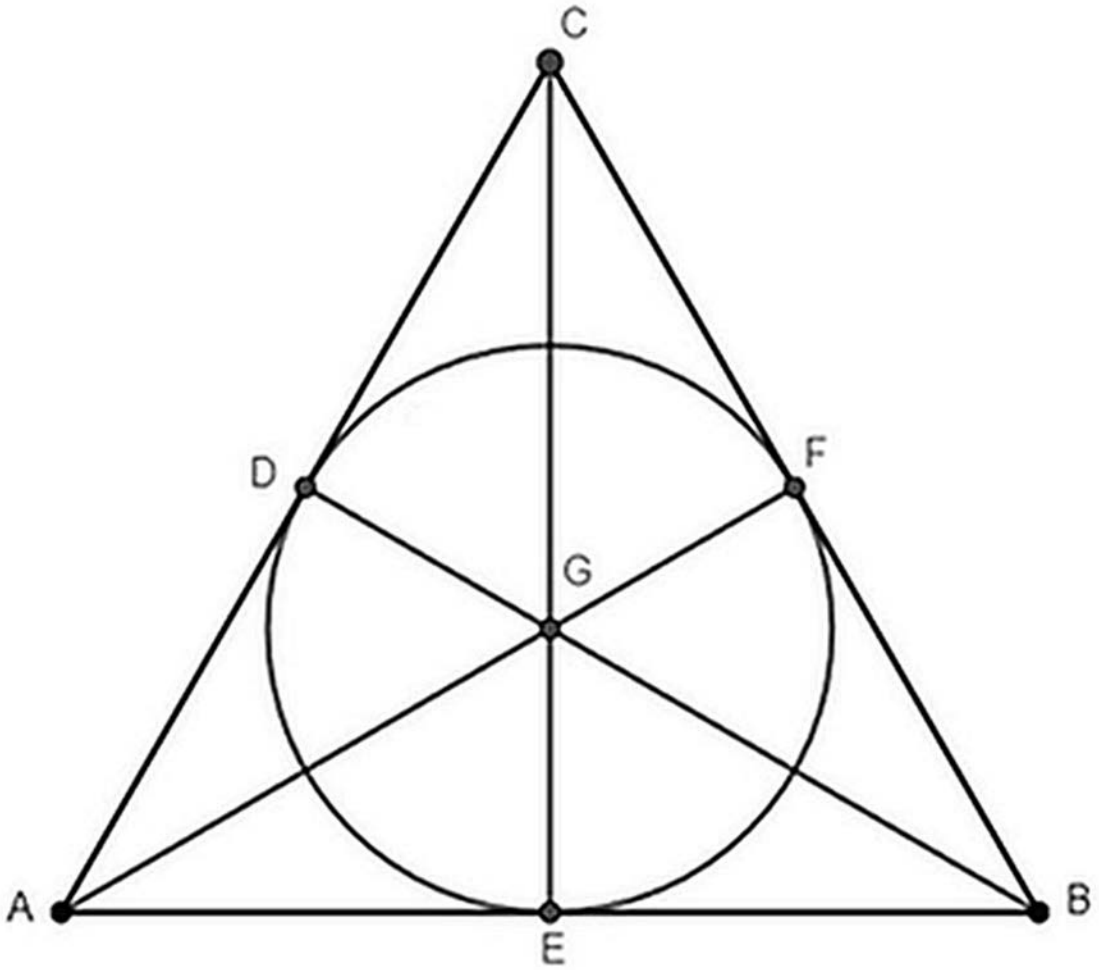
Teaching combinatorics is a vast and complex topic that can be approached differently depending on the student's school level, educational goals, and interests (Braithwaite & Goldstone, 2013; Vásquez et al., 2021; Tillema & Burch, 2022; Andriano et al., 2023). Of course, presenting the topic at the elementary school level is different from presenting it at the university level.

As we saw in the previous chapter, a first suggestion is to start with simple and concrete problems so as not to present the topic as a mountain to climb; In some cases, the reference to artifacts can greatly help. Secondly, it effectively demonstrates how combinatorics is connected to statistics, probability, algebra, and geometry, and more broadly, to all the different STEM areas. From a methodological point of view, it is necessary to encourage discovery instead of presenting formulas and rules "ready for use" to develop critical thinking skills through problem-solving activities, games and simulations with group and collaborative work (Mashiach Eizenberg & Zaslavsky, 2004; Lockwood, 2013; Lockwood, et al., 2020; Lockwood, 2022). Finally, the importance of educational and technological tools should not be underestimated, from simulation software to educational apps, educational videos on YouTube, and versatile programming languages (Logofătu & Gruber, 2014) such as Python. There are interesting examples of connecting some aspects of mathematics to combinatorics using the GeoGebra software and Python.

DOI: 10.4018/979-8-3373-3089-1.ch014

An example of how it is possible to tackle non-trivial problems using a methodology that connects “playful” situations to theory and programming is as follows: consider a game in which you move along a structure by throwing a die starting from the central point.

Figure 1. Pyramid game



Depending on the result, we move to one of the external vertices. The game continues with a new roll of the dice. If the same number as the node comes up, the game ends. To view how the game works, one can use the following program:

Program 140 Pyramid Game

```
import matplotlib.pyplot as plt
import numpy as np
import random
# Function to draw the triangle, the inscribed circle, and the nodes numbered from 1 to 6
```