TOWER OF HANOI

A PROJECT REPORT

Submitted by:

Taranjeet Singh (24MCA20073)

In partial fulfilment for the award of the degree of

Masters of Computer Applications

IN

University Institute of Computing

Chandigarh University

NH-95 Chandigarh-Ludhiana
Highway, Sahibzada Ajit
Singh Nagar (Mohali), Punjab 140413.

April 2025

ACKNOWLEDGMENT

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of person whose ceaseless cooperation made it possible whose constant guide and encouragement crown all efforts with success.

I am gratitude to my project guide Dr. Majid Bashir for the guidance, inspiration and constructive suggestion that helpful me in the preparation of my this mini project.

TABLE OF CONTENT

Overview

The Tower of Hanoi also called the Tower of Brahma or Lucas is a mathematical game or puzzle. It consists of three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

- Only one disk can be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
- No disk may be placed on top of a smaller disk.
- With three disks, the puzzle can be solved in seven moves. The minimum number of moves required to solve a Tower of Hanoi puzzle is 2n - 1, where n is the number of disks.

The puzzle can be played with any number of disks, although many toy versions have around seven to nine of them. The minimum number of moves required to solve a Tower of Hanoi puzzle is 2n - 1, where n is the number of disks.

Origin

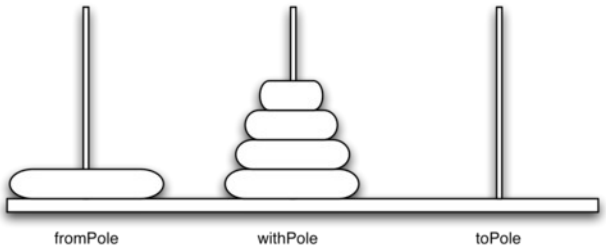
The puzzle was invented by the French mathematician Édouard Lucas in 1883. There is a story about an Indian temple in Kashi Vishwanath which contains a large room with three time-worn posts in it surrounded by 64 golden disks. Brahmin priests, acting out the command of an ancient prophecy, have been moving these disks, in accordance with the immutable rules of the Brahma, since that time. The puzzle is therefore also known as the Tower of Brahma puzzle. According to the legend, when the last move of the puzzle will be completed, the world will end. It is not clear whether Lucas invented this legend or was inspired by it.

If the legend were true, and if the priests were able to move disks at a rate of one per second, using the smallest number of moves, it would take them 264-1 seconds or roughly 585 billion years or 18,446,744,073,709,551,615 turns to finish, or about 127 times the current age of the sun.

Applications

1. The Tower of Hanoi is frequently used in psychological research on problem solving.
2. It is also used as a Backup rotation scheme when performing computer data Backups where multiple tapes/media are involved.
3. Tower of Hanoi is popular for teaching recursive algorithms to beginning programming students.
4. The Tower of Hanoi is also used as a test by neuropsychologists trying to evaluate frontal lobe deficits.

An Example Arrangement of Disks for the Tower of Hanoi



Here is a high-level outline of how to move a tower from the starting pole, to the goal pole, using an intermediate pole:

- Move a tower of height-1 to an intermediate pole, using the final pole.
- Move the remaining disk to the final pole.
- Move the tower of height-1 from the intermediate pole to the final pole using the original pole.

As long as we always obey the rule that the larger disks remain on the bottom of the stack, we can use the three steps above recursively, treating any larger disks as though they were not even there. The only thing missing from the outline above is the identification of a base case. The simplest Tower of Hanoi problem is a tower of one disk. In this case, we need move only a single disk to its final destination. A tower of one disk will be our base case.

Here is the algorithm to move the disc from source to destination.

```
def moveTower(height,fromPole, toPole, withPole):
if height >= 1:
      moveTower (height-1, fromPole,withPole,toPole)
moveDisk(fromPole,toPole)        moveTower(height-
1,withPole,toPole,fromPole) def moveDisk(fp,tp):
   print("moving disk from",fp,"to",tp)
moveTower(3,"A","B","C")
```

output :-

```
moving disk from A to B
moving disk from A to C
moving disk from B to C
moving disk from A to B
moving disk from C to A
moving disk from C to B
moving disk from A to B
```

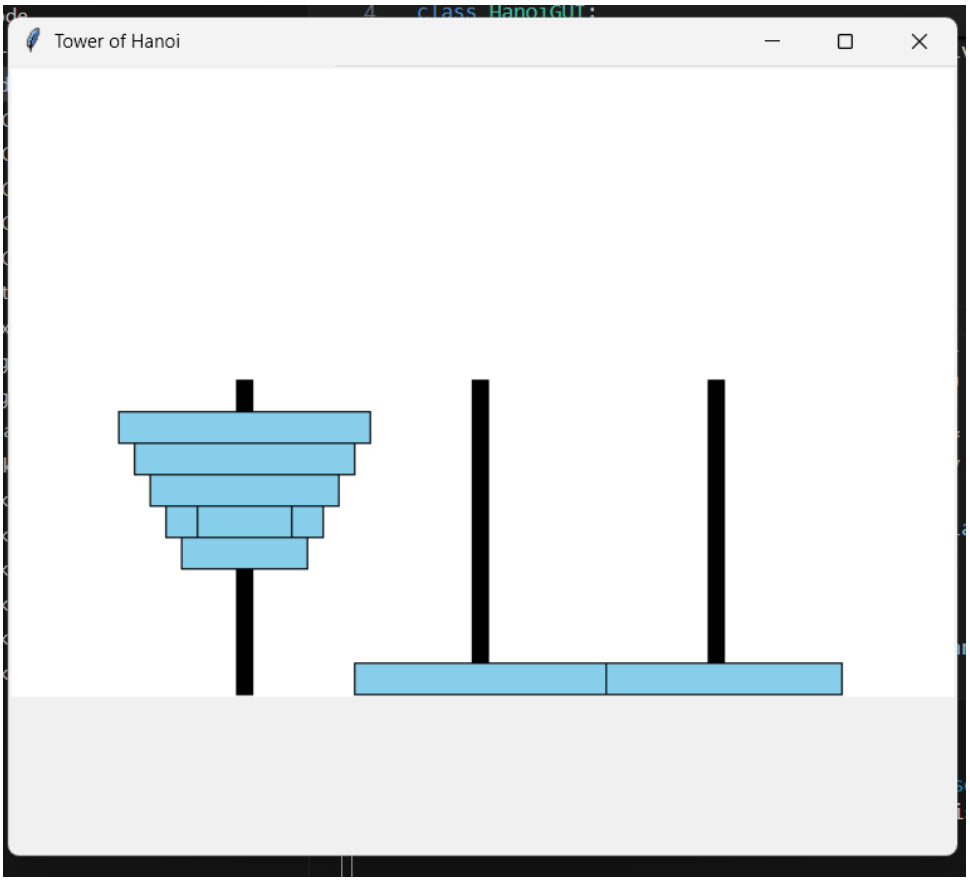
Source code of Tower of Hanoi

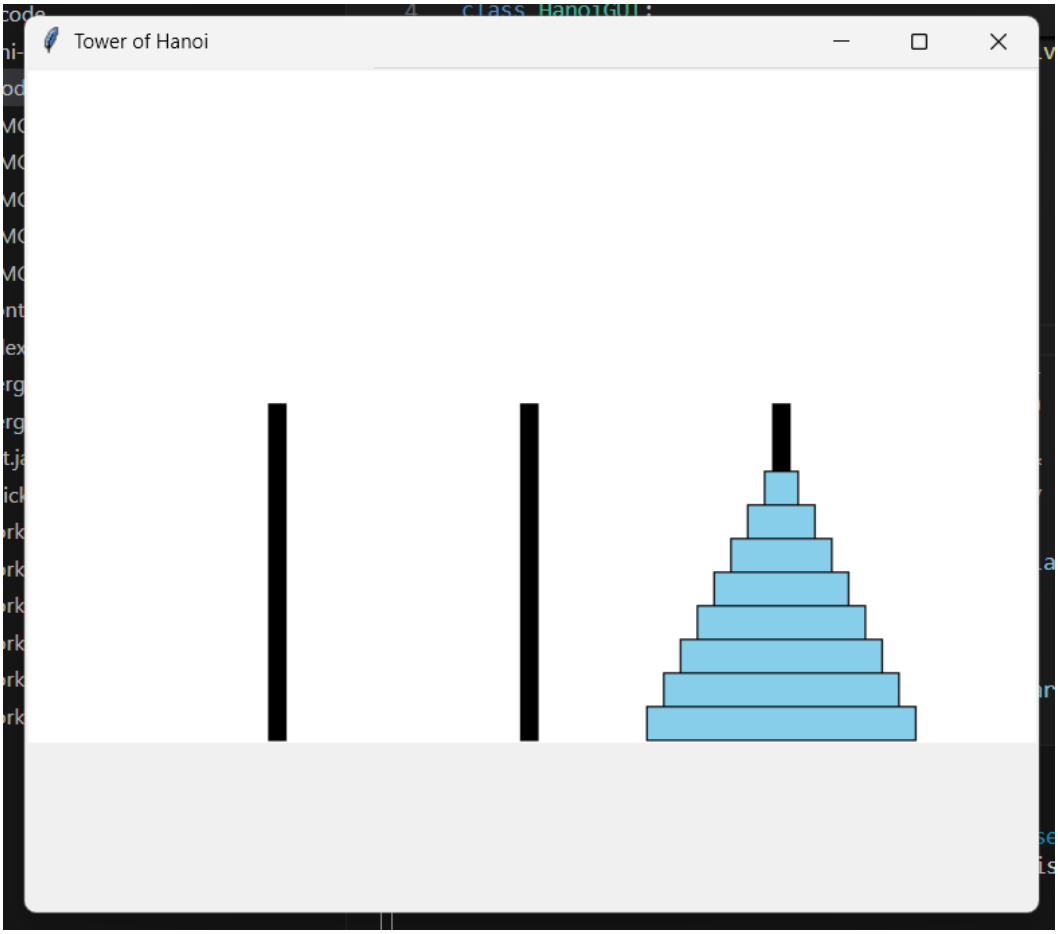
mini-project > Code.py

```python
import tkinter as tk
import time

class HanoiGUI:
    def __init__(self, root, num_disks):
        self.root = root
        self.root.title("Tower of Hanoi")

        self.canvas_width = 600
        self.canvas_height = 400
        self.canvas = tk.Canvas(root, width=self.canvas_width, height=self.canvas_height, bg="white")
        self.canvas.pack()

        self.num_disks = num_disks
        self.disk_height = 20
        self.peg_width = 10
        self.peg_height = 200
        self.peg_positions = [150, 300, 450]
        self.pegs = [[], [], []]

        # Create pegs
        for x in self.peg_positions:
            self.canvas.create_rectangle(x - self.peg_width // 2, self.canvas_height - self.peg_height,
                                         x + self.peg_width // 2, self.canvas_height, fill="black")

        # Create disks
        for i in range(num_disks, 0, -1):
            width = i * 20
            x = self.peg_positions[0]
            y = self.canvas_height - (num_disks - len(self.pegs[0])) * self.disk_height
            disk = self.canvas.create_rectangle(x - width // 2, y - self.disk_height,
                                                x + width // 2, y, fill="skyblue")
            self.pegs[0].append(disk)

        # Start solving
        self.root.after(500, lambda: self.solve(num_disks, 0, 2, 1))

    def move_disk(self, from_peg, to_peg):
        disk = self.pegs[from_peg].pop()
        self.pegs[to_peg].append(disk)

        # Update GUI
        self.canvas.update()
        self.canvas.after(500)

        disk_index = len(self.pegs[to_peg]) - 1
        width = (self.num_disks - disk_index) * 20
        x = self.peg_positions[to_peg]
        y = self.canvas_height - disk_index * self.disk_height
        self.canvas.coords(disk, x - width // 2, y - self.disk_height, x + width // 2, y)

    def solve(self, n, source, target, auxiliary):
        if n == 1:
            self.move_disk(source, target)
        else:
            self.solve(n - 1, source, auxiliary, target)
            self.move_disk(source, target)
            self.solve(n - 1, auxiliary, target, source)

# Run the GUI
if __name__ == "__main__":
    root = tk.Tk()
    num_disks = int(input("Enter number of disks: "))
    app = HanoiGUI(root, num_disks)
    root.mainloop()
```

Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\WINDOWS\System32\WindowsPowerShell\v1.0> & "C:/Users/Taranjeet Singh/AppData/Local/Programs/Python/Python313/python.exe" c:/OneDrive/Documents/
University_WorkMaterial/[MCA]/Semester_2/Design_Analysis_and_Algorithm/mini-project/Code.py
Enter number of disks: 8
```




Tower of Hanoi

Conclusion

From the proper analysis, this project tells that we can move the number of discs from source to destination rod with the help of one intermediate rod in such a way that smallest disc is placed at the top of the rod. And largest one is placed at the bottom, thus making a conical shape.

This is used in psychological research on problem solving.

This concept is also used in the development of the TURF framework for the representation of human computer Interaction.

Bibliography

Books :-

c++ programming by E.balagurusamy

Data structure by t.schaum series

Website :-
happycoding.com
Codingfox.co