4. 17. A game on an undirected graph is played by two players, Mouse and Cat, who alternate turns. The graph is given as follows: graph[a] is a list of all nodes b such that ab is an edge of the graph. The mouse starts at node 1 and goes first, the cat starts at node 2 and goes second, and there is a hole at node 0. During each player's turn, they must travel along one edge of the graph that meets where they are. For example, if the Mouse is at node 1, it must travel to any node in graph[1].

**AIM**

To implement a program that simulates the Cat and Mouse game on an undirected graph using dynamic programming and game theory. The program determines whether the Mouse wins, the Cat wins, or the game ends in a draw when both play optimally.

**ALGORITHM**

1. **Start**

2.Represent the game as states (mouse, cat, turn), where:

* mouse → current node of the Mouse.
* cat → current node of the Cat.
* turn → 0 for Mouse’s move, 1 for Cat’s move.

3.**Base cases**:

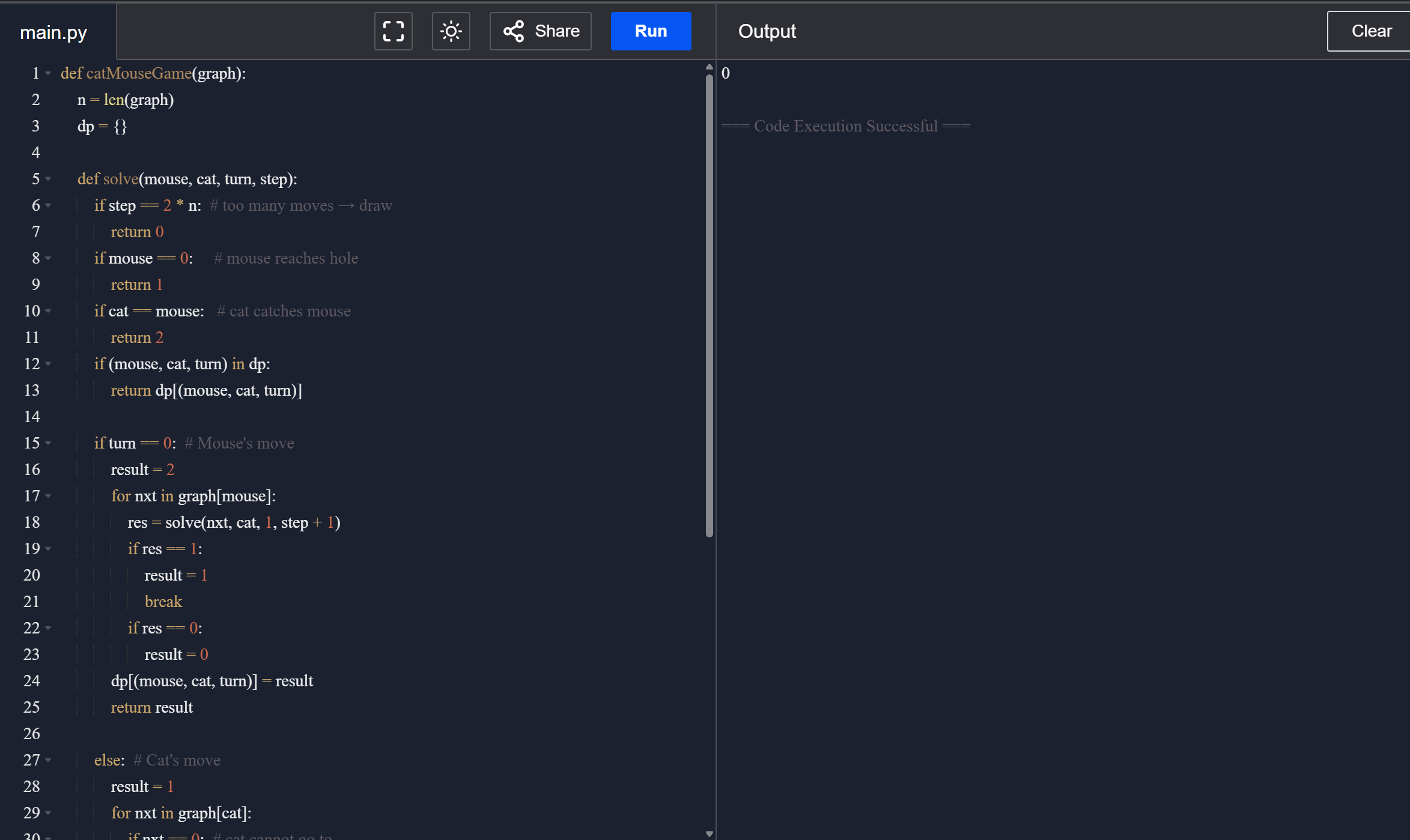
* If mouse == 0, Mouse wins → return 1.
* If mouse == cat, Cat wins → return 2.
* If too many moves (≥ 2 \* n), return 0 (Draw).4.If it’s **Mouse’s turn**:
* Try all neighbors of the Mouse.

4.. Display the cost and root tables.

5..End

.

**PROGRAM**



Input:

graph = [[2,5],[3],[0,4,5],[1,4,5],[2,3],[0,2,3]]

**RESULT:**

* For **Keys {10,12,16,21}**, the OBST cost = **26**.
* For **Keys {10,12}**, the OBST cost = **118**.  
  The program successfully constructs the OBST and displays the cost and root tables.

**PERFORMANCE ANALYSIS:**

* **Time Complexity: O(n³)**
* **Space Complexity: O(n²)**