

F. Prepare a table indicating the memory and time requirements to solve Puzzle-8 instances (depth “d”) using your graph search agent

Memory and Time requirements

Since we are using BFS

Space complexity: $O(b \cdot d)$

Time complexity: $O(b \cdot d)$

where b = branching factor & d = depth of state space

[illegible]

The screenshot shows a Java IDE with a project named 'Apples'. The 'Main.java' file is open, displaying a graph-based solution for a puzzle. The graph consists of nodes representing states (e.g., 'Apple 0', 'Apple 1', 'Apple 2') and edges representing transitions with associated costs (e.g., 'Cost of the movement: 8'). The code includes a 'main' method that initializes the graph and a 'search' method that finds the shortest path from the initial state to the goal state.

Number of transitions to get to the goal state = 9

Number of visited states = 1363

Total cost of solution = 47

Number of Nodes popped out of the queue = 843

Time taken = 149