**PROJECT REPORT**

1. **The definition of the problem**

**• States:** We get the number of the blocks on each part of our block world and the names of the blocks from the user. Then we let the user choose a goal block to keep all the blocks at the part he/she had chosen.

Graphical user interface, text

Description automatically generated

**• The Initial State:** Any state can be determined as the initial state.

Text

Description automatically generated

**• Actions:** We evaluated every possible movement case for picking up and putting down blocks in positions p, q, or r.

Text

Description automatically generated

**• The Transition Model:** If the stack action is taken, the block being stacked is placed on top of the other block, and its position on the surface is updated accordingly. Similarly, if the pick-up action is taken, the block is removed from its previous position on the surface, and its state is updated to indicate that it is now being held.

Text

Description automatically generated

**• Node:**

Text

Description automatically generated

• **Queue:**

Graphical user interface, text

Description automatically generated

**• Goal Test:** The goal test of the Block World problem checks whether the current state matches the desired end state where the blocks are arranged in a specific configuration on the surface.

Text

Description automatically generated

1. **The definition of the heuristic function that will be used:**

Text

Description automatically generated

This code implements a heuristic function for a search algorithm that estimates the cost of reaching the goal state from the current state in a problem-solving domain involving sorting position arrays. It counts the number of out-of-order positions in the current and goal states and returns the difference between them as a float. The heuristic function is used by a search algorithm to prioritize exploring states closer to the goal.

1. **As a result of performing several simulations, the results obtained for each algorithm**

* **Breadth-First Search:**

Text

Description automatically generated

* **Uniform-Cost Search:**

Text

Description automatically generated

* **Depth-First Search:**

Text

Description automatically generated

* **Depth-Limited Search:**

Text

Description automatically generated

* **Iterative Deepening Depth-First Search:**

Text

Description automatically generated

* **Greedy Best-First Search:**

Text

Description automatically generated

* **A\* Search:**

Text

Description automatically generated