BICOL UNIVERSITY COLLEGE OF SCIENCE

CS Elective – Artificial Intelligence

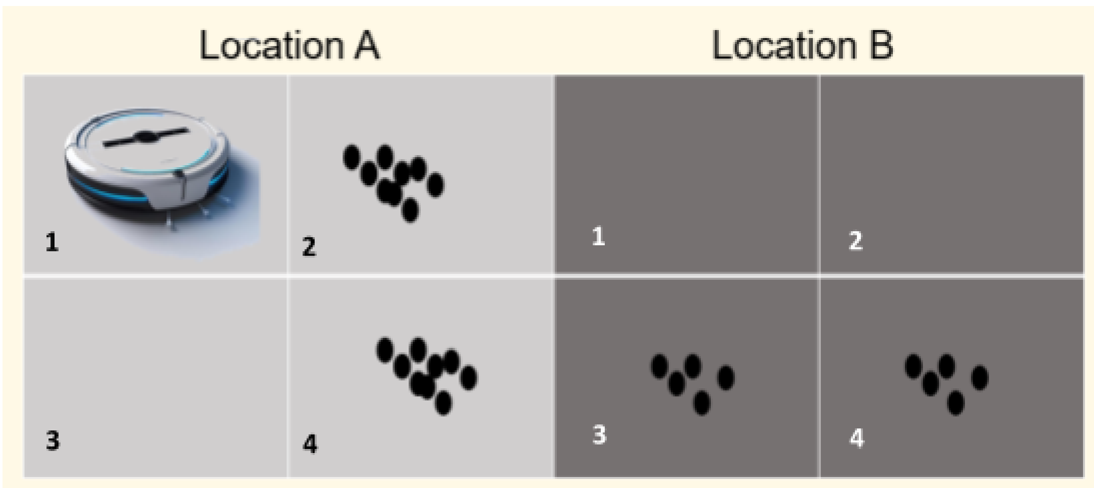
Class Participation #1

CANONIZADO, MICHAEL XAVIER, E.

PART I. AI Agents

1. **Vacuum World Problem. Path cost**: Each step costs 1. What is the total number of actions to reach the goal? Show the paths to justify your answer. 10 points

e.g. A1 -> A2-> SIPHON -> …

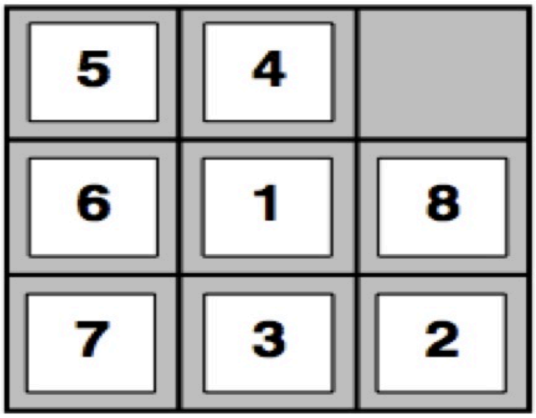


Optimal Paths:

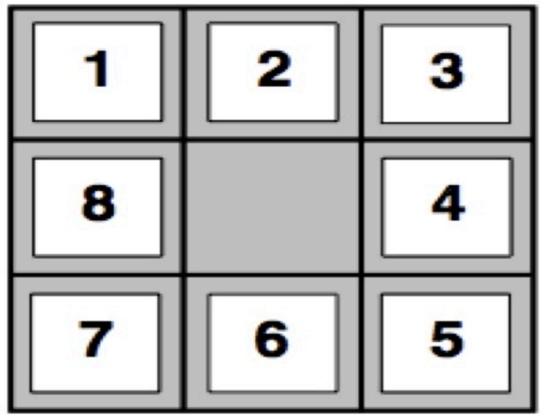
1. A1 -> A2 -> SIPHON -> B1 -> B2 -> B4 -> SIPHON -> B3 -> SIPHON -> A4 -> SIPHON -> A3
2. A1 -> A3 -> A4 -> SIPHON -> A2 -> SIPHON -> B1 -> B3 -> SIPHON -> B4 -> SIPHON -> B2
3. A1 -> A3 -> A4 -> SIPHON -> A2 -> SIPHON -> B1 -> B2 -> B4 -> SIPHON -> B3 -> SIPHON
4. A1 -> A3 -> A4 -> SIPHON -> B3 -> SIPHON -> B4 -> SIPHON -> B2 -> B1 -> A2 -> SIPHON

   Where all paths cost 11 each (1 move, 1 siphon). This is the optimal path for the vacuum which follows a Hamiltonian path, where the vacuum must visit each tile exactly once. This is the most optimal approach since the vacuum is doing uninformed search, and other uninformed searching algorithms like DFS or BFS, may revisit tiles and therefore do more than 7 moves without siphon, or 11 cost with siphon.

2. **8 Puzzle Problem. Path cost**: Each step costs 1. What is the total number of actions to reach the goal? Show the paths to justify your answer. 10 points



Start State



Goal State

After trying for hours, the closest I got was:
[4R, 1U,8L, 2U, 3R, 8D, 6R, 5D, 1L, 4L, 2U, 6R, 5R, 7U, 8L, 5D, 6L, 3U, 5R, 6D, 4D, 2L, 3U, 4R]
giving the result:

[1 , 2 , 3]

[7 ,   , 4]

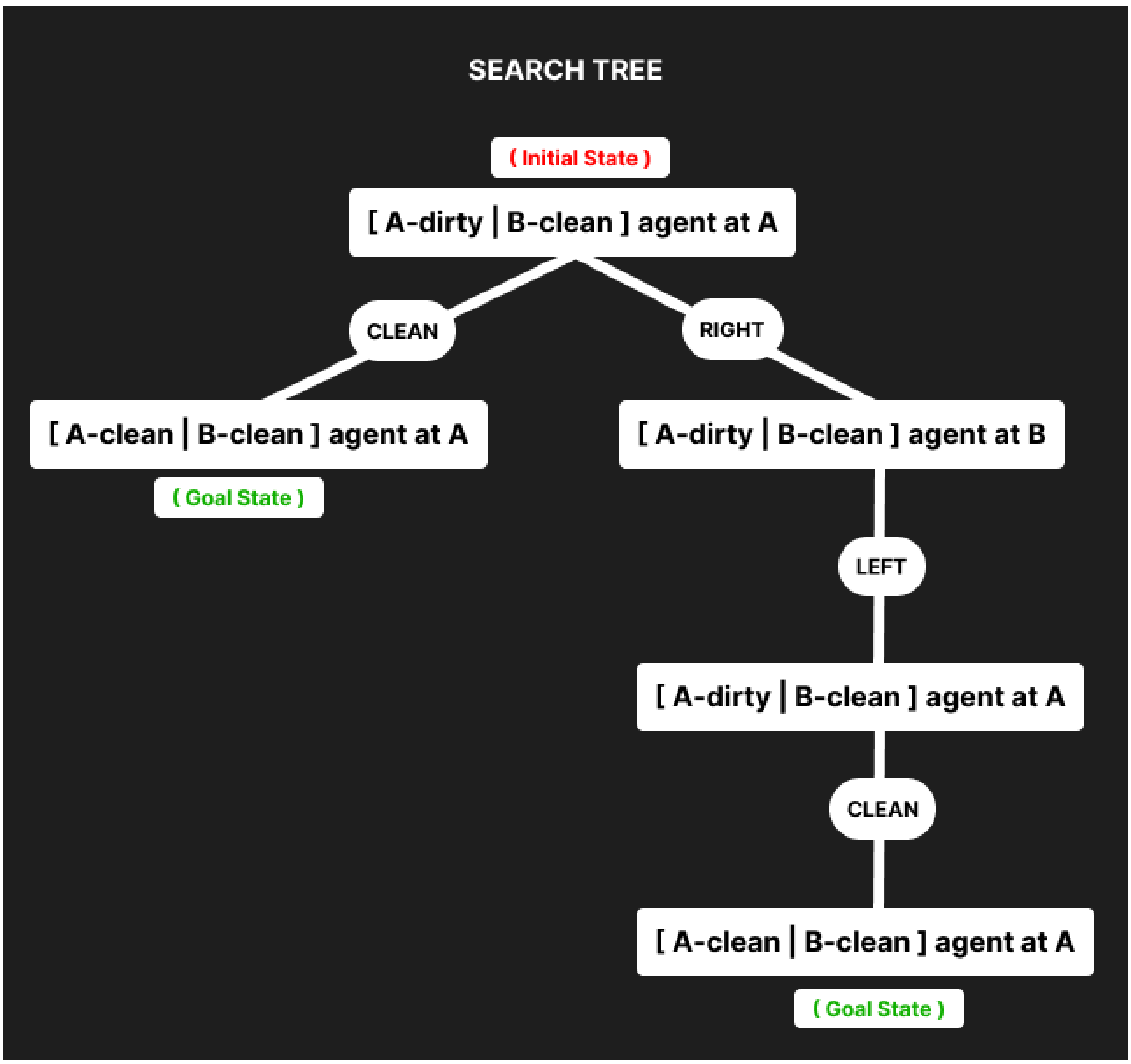
[8 , 6 , 5]

I then searched online if the goal state was achievable, and after researching, I found that it was impossible to reach the goal state.

Part II. Uninformed vs. Informed Search (2-Room Vacuum World)

Scenario Setup

- Environment: 2 **rooms in a row** → A – B
- Initial state:
  - Agent starts at **Room A**
  - Room A = **dirty**
  - Room B = **clean**
- Goal: All rooms are clean.
- Actions: **Clean, Move Left, Move Right** (cost = 1 per action).
- Requirements: Apply all Uninformed and Informed Search Algorithms (show all expansions, including non-optimal path).



1) **Breath First Search:**

Level 1:

1) Clean A -> [A - clean , B - clean] , Agent at A | **GOAL FOUND**

2) Move right -> [A - dirty , B - clean] , Agent at B

Moves: [Clean]

2) **Depth First Search:**

Level 1:

1) Clean A -> [A - clean , B - clean] , Agent at A | **GOAL FOUND**

Moves: [Clean]

3) **Uniform Cost Search:**

Level 1:

1) Clean A -> [A - clean , B - clean] , Agent at A | (Cost = 1)| **GOAL FOUND**

2) Move Right -> [A - dirty , B - clean] , Agent at B | (Cost = 1)

Moves: [Clean]

4) **Depth Limited Search (limit = 2):**

Level 1:

1) Clean A -> [A - clean , B - clean] , Agent at A | **GOAL FOUND**

Moves: [Clean]

5) Iterative Deeping Search

Iteration 0:

1) Check Root -> [A - dirty , B - clean] , Agent at A | Not Goal

Iteration 1:

1) Clean A -> [A - clean , B - clean] , Agent at A | (Cost = 1)| **GOAL FOUND**

2) Move Right -> [A - dirty , B - clean] , Agent at B | (Cost = 1) | Not Goal

Moves: [Clean]

6) Greedy Best First Search [h(n) = num of dirty rooms]:

Start:

1) [A - dirty , B - clean] , Agent at A | h = 1

Expansion 1:

1) Clean -> [A - clean, B - clean] , Agent at A | h = 0
2) Move right -> [A - dirty , B - clean] , Agent at B | h = 1

GBFS sees that h(n) reached 0 therefore chooses that path

Moves: [Clean]

7) A* Search [f(n) = g(n) + h(n) | f(n) = predicted total cost, g(n) = cost so far, h(n) = num of dirty rooms]:

Initial:

1) [A - dirty , B - clean] , Agent at A | f = 1, g = 0, h = 1

Start:

1) Clean -> [A - clean, B - clean] , Agent at A | f = 1, g = 1, h = 0
2) Move Right -> [A - dirty, B - clean] , Agent at B | f = 2, g = 1, h = 1

A sees that h = 0, which indicates the goal state has been reached, and that the other path leads to more cost but no h didn't change*

Moves: [Clean]