BICOL UNIVERSITY COLLEGE OF SCIENCE

CS Elective – Artificial Intelligence

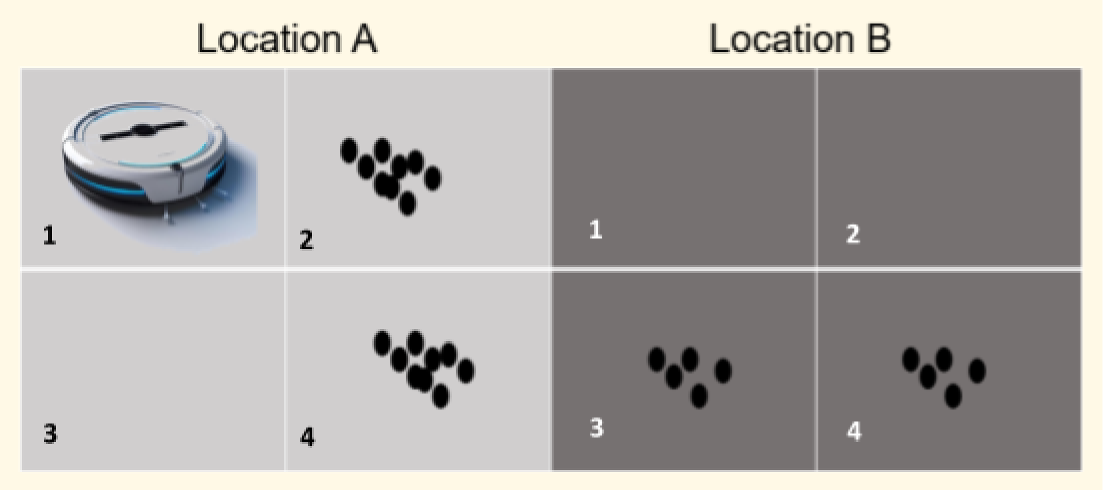
Class Participation #1

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 -> …

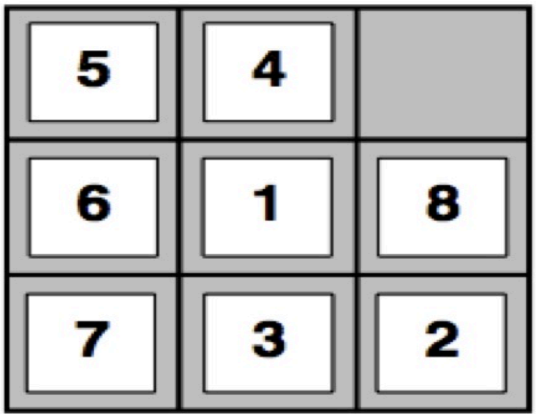


Optimal Paths:

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

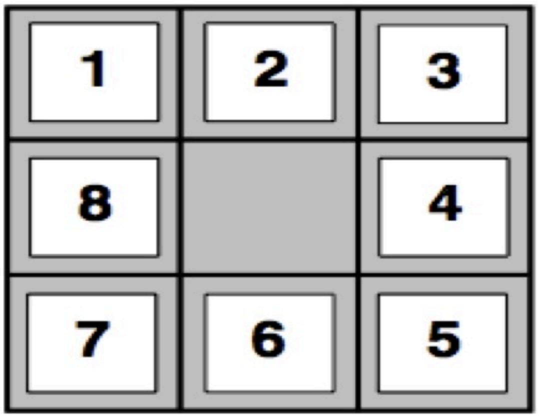
Where all paths cost 11 each (1 move, 1 siphon). This is the optimal path for the vacuum which follows a Hamiltonian path, where the vacuum must visit each tile exactly once. This is the most optimal approach since the vacuum is doing uninformed search, and other uninformed searching algorithms like 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



5	4	
6	1	8
7	3	2

Start State



1	2	3
8		4
7	6	5

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]

We can verify if a goal state is achievable by checking whether its inversions are even or odd:

Flattened Goal State: 1, 2, 3, 8, 4, 7, 6, 5

- **1:** - 1 > 2 = N - 1 > 3 = N - 1 > 8 = N - 1 > 4 = N - 1 > 7 = N - 1 > 6 = N - 1 > 5 = N	- **2:** - 2 > 3 = N - 2 > 8 = N - 2 > 4 = N - 2 > 7 = N - 2 > 6 = N - 2 > 5 = N	- **3:** - 3 > 8 = N - 3 > 4 = N - 3 > 7 = N - 3 > 6 = N - 3 > 5 = N	- **8:** - 8 > 4 = Y - 8 > 7 = Y - 8 > 6 = Y - 8 > 5 = Y
- **4:** - 4 > 7 = N - 4 > 6 = N	- **7:** - 7 > 6 = Y - 7 > 5 = Y	- **6:** - 6 > 5 = Y	- **5:** - None

- 4 > 5 = N			

Total inversions for the goal state: **4 + 2 + 1 = 7**

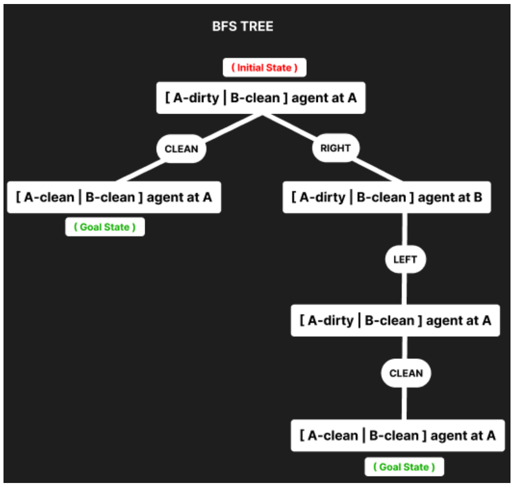
Since 7 is an odd number, there is no solution for this 8 puzzle goal state.

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

Scenario Setup

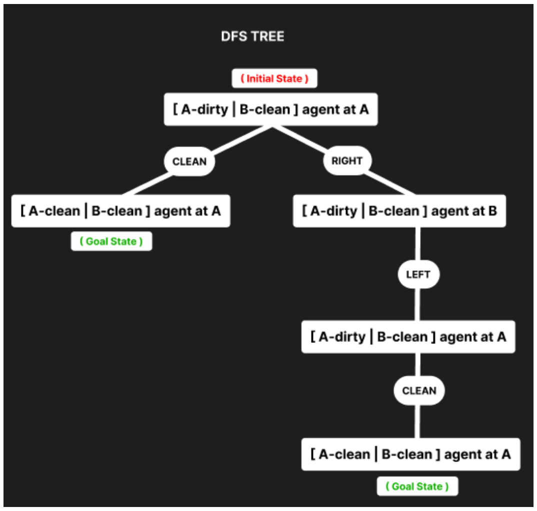
- Environment: 2 **rooms in a row** ⟶ A – B
- Initial state:
  - Agent starts at **Room A**
  - Room A = **dirty**
  - Room B = **clean**
- Goal: All rooms are clean.
- Actions: **Clean, Move Left, Move Right** (cost = 1 per action).
- Requirements: Apply all Uninformed and Informed Search Algorithms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

1) Breath First Search:


BFS TREE
(Initial State)
[A-dirty | B-clean] agent at A
CLEAN
RIGHT
[A-clean | B-clean] agent at A
(Goal State)
[A-dirty | B-clean] agent at B
LEFT
[A-dirty | B-clean] agent at A
CLEAN
[A-clean | B-clean] agent at A
(Goal State)

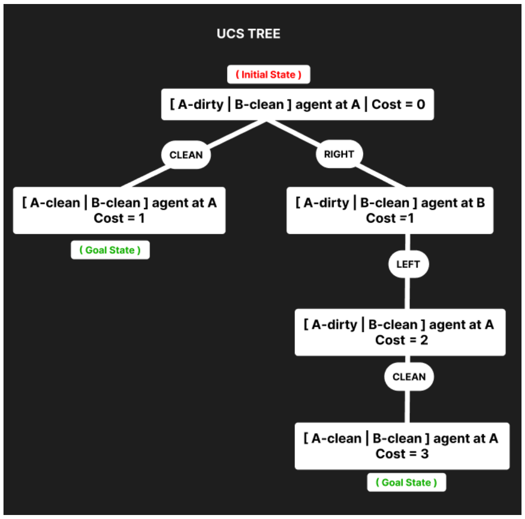
Moves: [Clean]

2) Depth First Search:


DFS TREE
(Initial State)
[A-dirty | B-clean] agent at A
CLEAN
RIGHT
[A-clean | B-clean] agent at A
(Goal State)
[A-dirty | B-clean] agent at B
LEFT
[A-dirty | B-clean] agent at A
CLEAN
[A-clean | B-clean] agent at A
(Goal State)

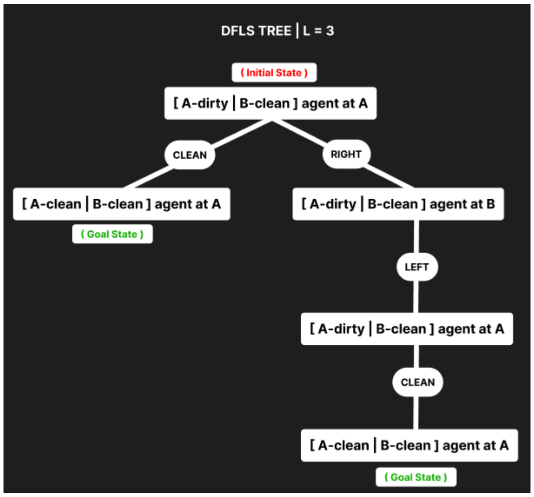
Moves: [Clean]

3) Uniform Cost Search:


UCS TREE
(Initial State)
[A-dirty | B-clean] agent at A | Cost = 0
CLEAN
RIGHT
[A-clean | B-clean] agent at A
Cost = 1
(Goal State)
[A-dirty | B-clean] agent at B
Cost =1
LEFT
[A-dirty | B-clean] agent at A
Cost = 2
CLEAN
[A-clean | B-clean] agent at A
Cost = 3
(Goal State)

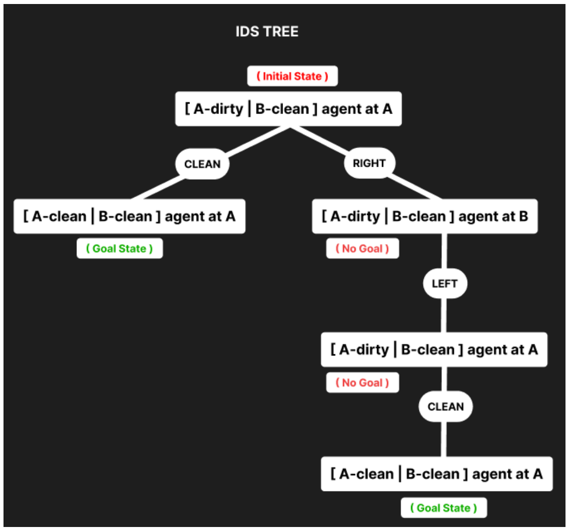
Moves: [Clean]

4) Depth Limited Search (limit = 2):


DFLS TREE | L = 3
(Initial State)
[A-dirty | B-clean] agent at A
CLEAN
RIGHT
[A-clean | B-clean] agent at A
(Goal State)
[A-dirty | B-clean] agent at B
LEFT
[A-dirty | B-clean] agent at A
CLEAN
[A-clean | B-clean] agent at A
(Goal State)

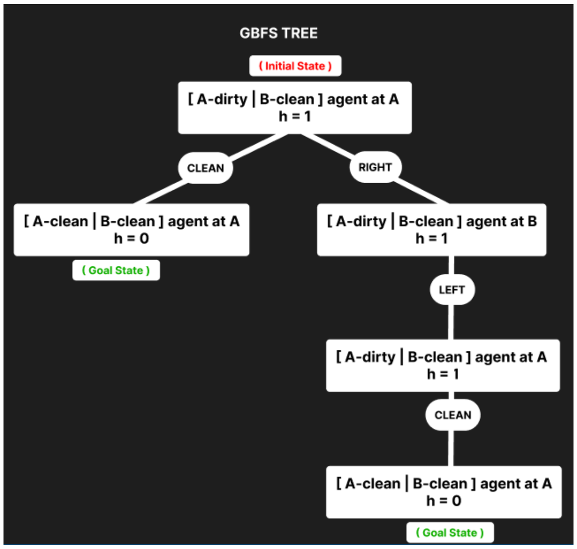
Moves: [Clean]

5) Iterative Deeping Search


IDS TREE
(Initial State)
[A-dirty | B-clean] agent at A
CLEAN
RIGHT
[A-clean | B-clean] agent at A
(Goal State)
[A-dirty | B-clean] agent at B
(No Goal)
LEFT
[A-dirty | B-clean] agent at A
(No Goal)
CLEAN
[A-clean | B-clean] agent at A
(Goal State)

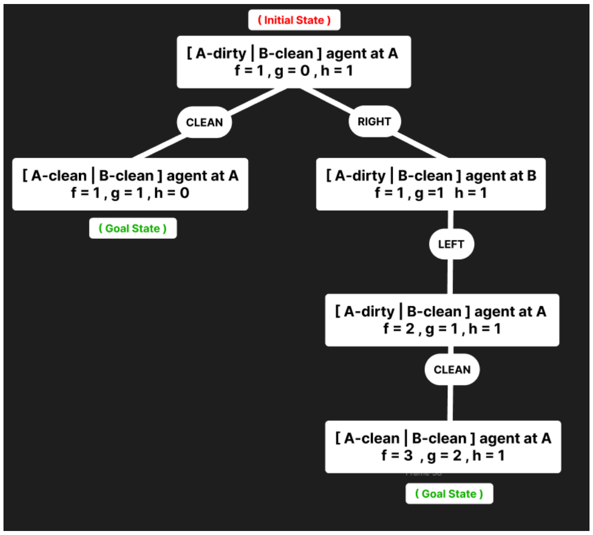
Moves: [Clean]

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


GBFS TREE
(Initial State)
[A-dirty | B-clean] agent at A
h = 1
CLEAN
RIGHT
[A-clean | B-clean] agent at A
h = 0
(Goal State)
[A-dirty | B-clean] agent at B
h = 1
LEFT
[A-dirty | B-clean] agent at A
h = 1
CLEAN
[A-clean | B-clean] agent at A
h = 0
(Goal State)

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]:



A sees that h = 0 when [Clean], 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]