Artificial Intelligence
Lab 2 Report

Vibhuti Ramn - 180010040
Vipul Nikam - 180010041

Command to run:

python3 180010040_180010041_Lab2.py

For selecting desired heuristic functions different heuristic functions are written in code so just enter values for heuristic function numbers from **1, 2 or 3** and output will get printed.

1. Brief description about domain::

Our task here is to simulate BFS and Hill Climbing in state space. It is based on the Blocks-World Domain problem. There are 6 blocks named from A to F. Starting with an initial state consisting of a fixed number of blocks arranged in 3 lists and it can move only top blocks of stacks and to achieve a target state which is a particular arrangement . of blocks by moving these blocks. Blocks World is a planning problem where goal state in advance and path to goal state is more important.

a. State space:

We had multiple states that depend on arrangement of blocks. A state in submitted code is indicated by a list indicating the block in which a particular block is placed. If a block is placed on a table, it is indicated by a space ' '. it can be moved a single block at a time in which there is no other block at same time and can be placed on any other block that is on top or on table.

b. Start node and goal node:

Start state	EBF	DA	C
Goal state	ADB	EFC	

c. MOVEGEN and GOAL TEST algorithm:

"It generates all possible movements that can be generated from the current position. to obtain states taking into account possible combinations of blocks. It is obtained by moving only one block on top and placing it where it is allowed. Then using heuristic function values to get the best neighbor. As already mentioned earlier the target state by its respective list of names. Therefore, we directly compare the heuristic value of the current state with the heuristic value of the target state. And if they match, thereafter, the current state is the target state.

2. Heuristic functions considered (minimum of 2):

1. Here we increment the heuristic value by one, if the given block is on the correct block/table. And we decrease value by one if it is on an incorrect block. This is a local heuristic function as it only checks local space of a block.
2. The 2nd heuristic function does not consider the block on which block rests under consideration, but the entire stack of blocks. Therefore, we will award one point for each of the blocks in the entire correct stack. If a stack contains even one misplaced block, we will give a negative point for each of the blocks in the entire stack.

Start state												Total
E	-1	B	-2	F	-3	D	-1	A	-2	C	-1	-10
Goal state												Total
A	1	D	2	B	3	E	1	F	2	C	3	12

3. This heuristic is almost the same as the 2nd one. In this we decrease value in the same way as the 2nd heuristic. But we give positive 1 to each block if it is placed correctly.

3. Best First search analysis and observation:

BFS uses two sets. One is an open dataset (generated but not selected yet) and the other is a closed dataset (already selected). Find the first one by selecting the node with the best heuristic value from the open list for expansion. The node is removed from the open list and each of its children is generated. For each generated child node, the best search first checks the open and closed lists to ensure that the first search is not a duplicate of the generated node or, if it is, a better way than the previously created duplicate. If so, it will be added to the closed list and a new node will be selected for the extension. This continues until the destination node is selected for the extension. The best first search for the least limited domain is complete. Scan all nodes before reporting failures. The best first discovery rate depends on the precision of the heuristic functions. The complexity of space and the complexity of time depend on the precision of the heuristic function. With good heuristics, the search will show the target for a long period of time. The time complexity of the first best search is less than the first wide search. It is more efficient than BFS and DFS. Best First Search allows you to switch between the ways the first search finds in width and space. Because the room is good first because the solution is found without counting all nodes, and the first search is good because it is not stuck in dead ends.

4. Hill Climbing and Best First search comparison in terms of:

a. States explored

b. Time taken

c. Reaching optimal solution.

BFS scans all nodes before reporting failures. It is complete but not optimal. You can bounce back if you get stuck at local lows / highs. Hill climbing expands nodes closest to the goal based on heuristic value. It does not keep history, algorithms cannot recover from failures in its policy. A major problem with escalation is its tendency to get stuck in local data. It

is neither optimal nor complete. It can be very fast if we use good heuristic functions. Only one item is taken into account at a time. Takes less space than best first search.

Heuristic	Algorithm	States Explored	Time taken	Reached goal/optimal state
1	BFS	36	0.0076	Yes if rejected by greedy state when already visited
		36		No if don't store history of visited nodes by greedy When gets stuck in a loop, without repeated checking. If used naïve greedy method
	Hill Climbing	4	0.0027	No. Get stuck at local optima.
2	BFS	6	0.0077	Yes
	Hill Climbing	6	0.0024	Yes
3	BFS	7	0.0082	Yes
	Hill Climbing	6	0.0032	Yes

XXX