Solve the following block world problem by finding (using Iterative Deepening Search – IDS) a sequence of steps to reach the given goal state from the given initial state. Your code should implement the following API.

- InitialState(): Returns the initial state of the problem.
- GoalTest(s): Returns a Boolean, "true" iff state s is a goal state.
- Cost(a): Returns the cost of action a.
- Actions(s): Returns the set of actions that are applicable to state s.
- ChildState(s, a): Requires that action a is applicable to state s, i.e., there is a transition $s \xrightarrow{a} s'$. Returns the outcome state s'.
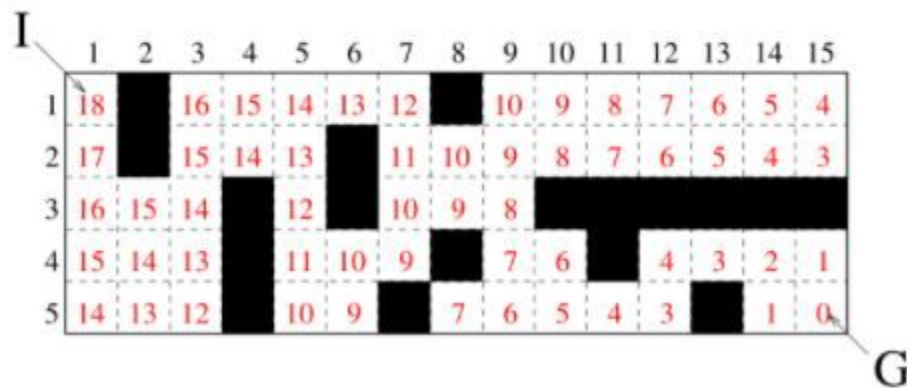
Initial State   Goal State
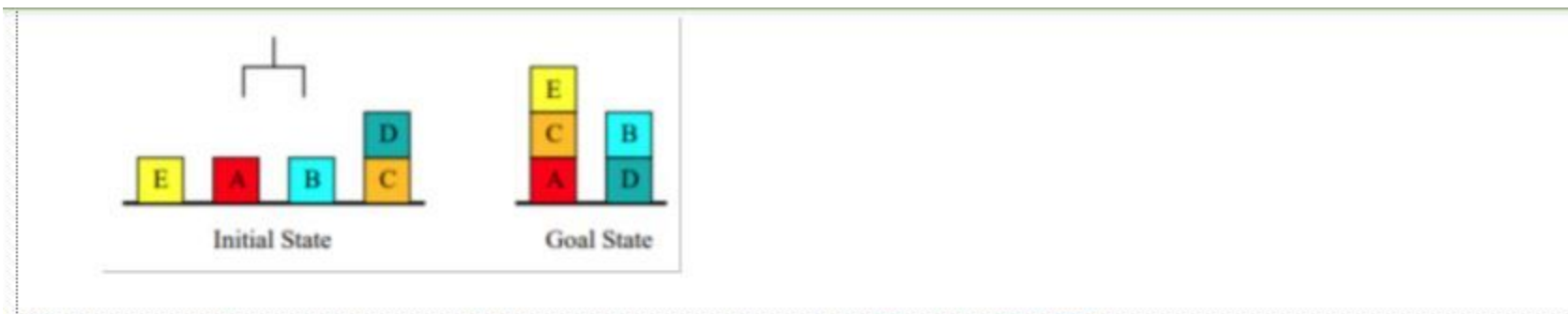
Solving the following grid-based path finding problem using

1. greedy best first search (h only)
2. uniform cost search (g only)
3. A* search (g+h)
4. IDA* search (g+h)

Manhattan Distance, "accurate h":

I	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	18	■	16	15	14	13	12	■	10	9	8	7	6	5	4
2	17	■	15	14	13	■	11	10	9	8	7	6	5	4	3
3	16	15	14	■	12	■	10	9	8	■	■	■	■	■	■
4	15	14	13	■	11	10	9	■	7	6	■	4	3	2	1
5	14	13	12	■	10	9	■	7	6	5	4	3	■	1	0

G

Consider a puzzle having a 4x4 square of numbers. There are four operations, each of which involves rotating the numbers in a 3x3 square clockwise. So, in the diagram below, one move is the cycle (1,2,3,7,11,10,9,5). Design the problem in PDDL so that starting from a random configuration one can reach the following goal configuration. The numbers maintain orientation-- they don't rotate; if they did, how would you design that in PDDL.

1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

Write a local search algorithm in any programming language to color the Divisions of Bangladesh in the following map using minimum number of colors so that adjacent divisions are assigned with different colors.

📕 Minimax with Alpha Beta Assignment ✅

📕 [Implement Tic tac toe](#) ✅

Problem 1 will carry 5 marks and Problem 2 will carry extra 5 marks.

Your progress ❓