TUTORIAL 2 : To understand State Space problem formulation

NAME : Ajay Kumar Prasad

CLASS : B.E. IT

ROLL NO. 49

SUBJECT : IS Lab

DOP	DOC	MARKS	SIGN

TUTORIAL 2: To understand State Space problem formulation

* AIM: To understand State Space based problem formulation for AI problems so that Problem Solving Agent can be applied.

* THEORY:

First we understand the problem solving agent. Agent first formulates goal and problem, then determines or rather searches an action sequence, after which it returns the next action to be executed in a sequential manner.

Defining the problem is referred to as Problem Formulation. It involves defining five things:

Initial State: It is the starting state that the problem is in

Actions: It defines all possible actions available to the agent, given it is in some state s currently. It is a function Action(s) that returns list of all possible actions.

Transition Model: also known as successor function which define which state is the system tend to move to when a particular action is executed by the agent. Successive application of transition model gives rise to what is known as state space

Problem Solving Agent Architecture

function SIMPLE-PROBLEM-SOLVING-AGENT(percept) returns an action

static: seq, an action sequence, initially empty

state, some description of the current world state

goal, a goal, initially null

problem, a problem formulation

state ← UPDATE-STATE(state, percept)

if seq is empty then do

goal ← FORMULATE-GOAL(state)

problem ← FORMULATE-PROBLEM(state, goal)

seq ← SEARCH(problem)

action ← FIRST(seq)

seq ← REST(seq)

return action

Goal Test: This act as a stopping condition when the state passed to this function is goal state it will return true and searching would stop.

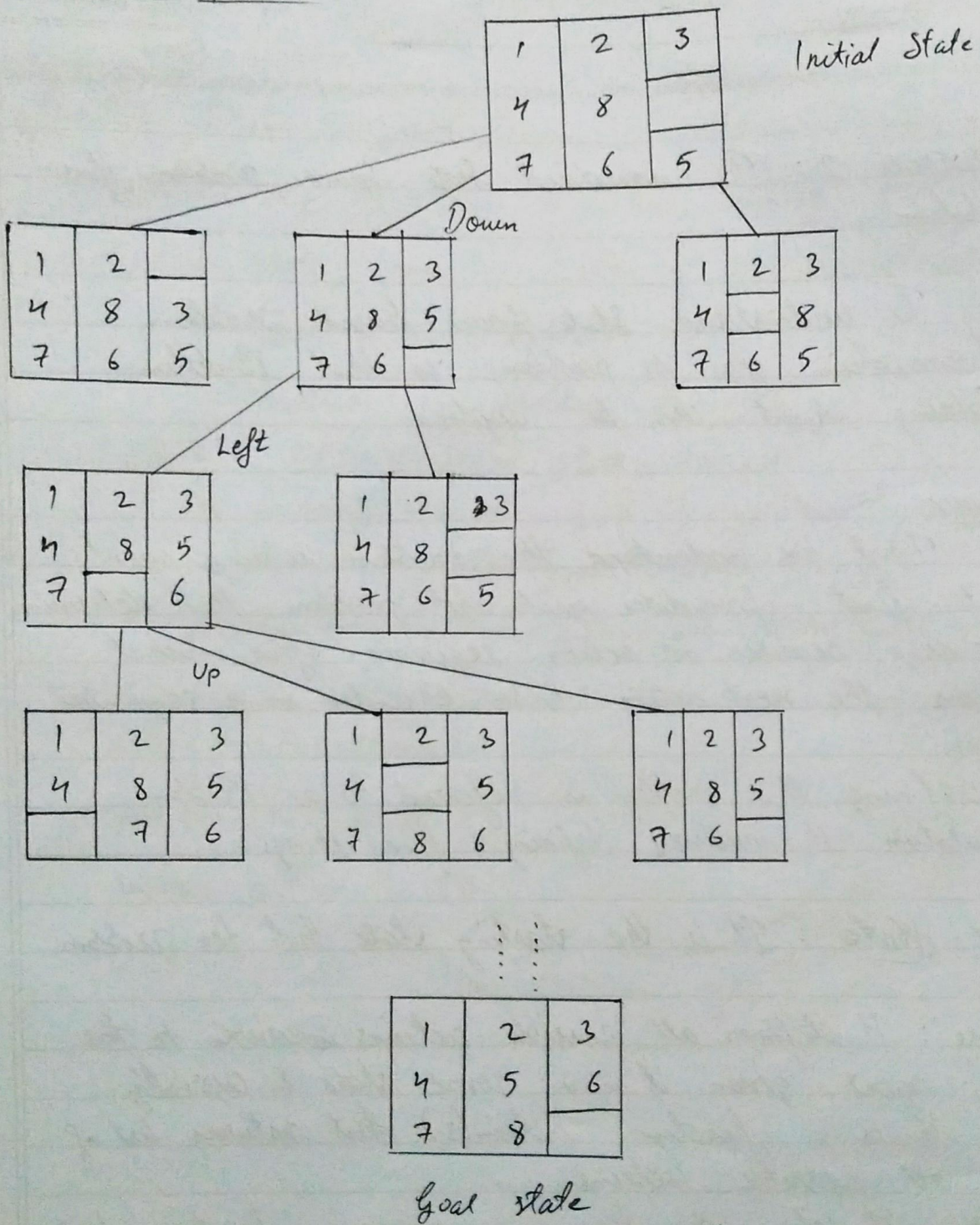
Path cost: It is accumulated cost of performing certain sequence of actions. This can be help in determining whether the action sequence under consideration is optimal.

Thus a problem can formally specified by identifying initial state, actuators, transition model, goal test and path cost. In terms of problem solving agent solution is the path from initial state to a goal state. optimal solution is the lowest path cost of all solutions.

* WORKING: Based on understanding of problem formulation students need to formulate following problems. This will clearly show state space up to depth level 3 or till goal node which ever is shallowest.

(i) 8-puzzle problem
The problem can be formulated as:
- States: States can be represented by a 3×3 matrix data structure with blank denoted by an underscore '_'.
1. Initial State: {{1, 2, 3}, {4, 8, _}, {7, 6, 5}}
2. Actions: The blank space moves in left, right, up, and down direction specifying the actions.
3. Successor function: If we apply 'down' operation, to the

8 puzzle problem (State Space)



start state, the next state has '5' and '-' switched.

4 Goal test: {{1,2,3},{4,5,6},{7,8,-}}

5 Path cost: No. of steps to reach to the final state.

- Solution:

{{1,2,3},{4,8,-},{7,6,5}} → {{1,2,3},{4,8,5},{7,6,-}}

{{1,2,3},{4,8,5},{7,-,6}} → {{1,2,3},{4,-,5},{7,8,6}}

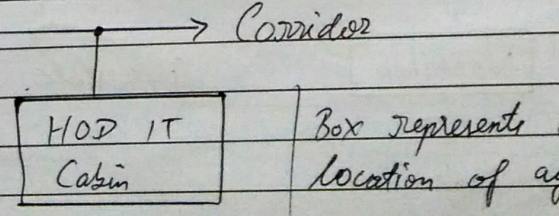
{{1,2,3},{4,5,-},{7,8,6}} → {{1,2,3},{4,5,6},{7,8,-}}

Path cost = 5 steps

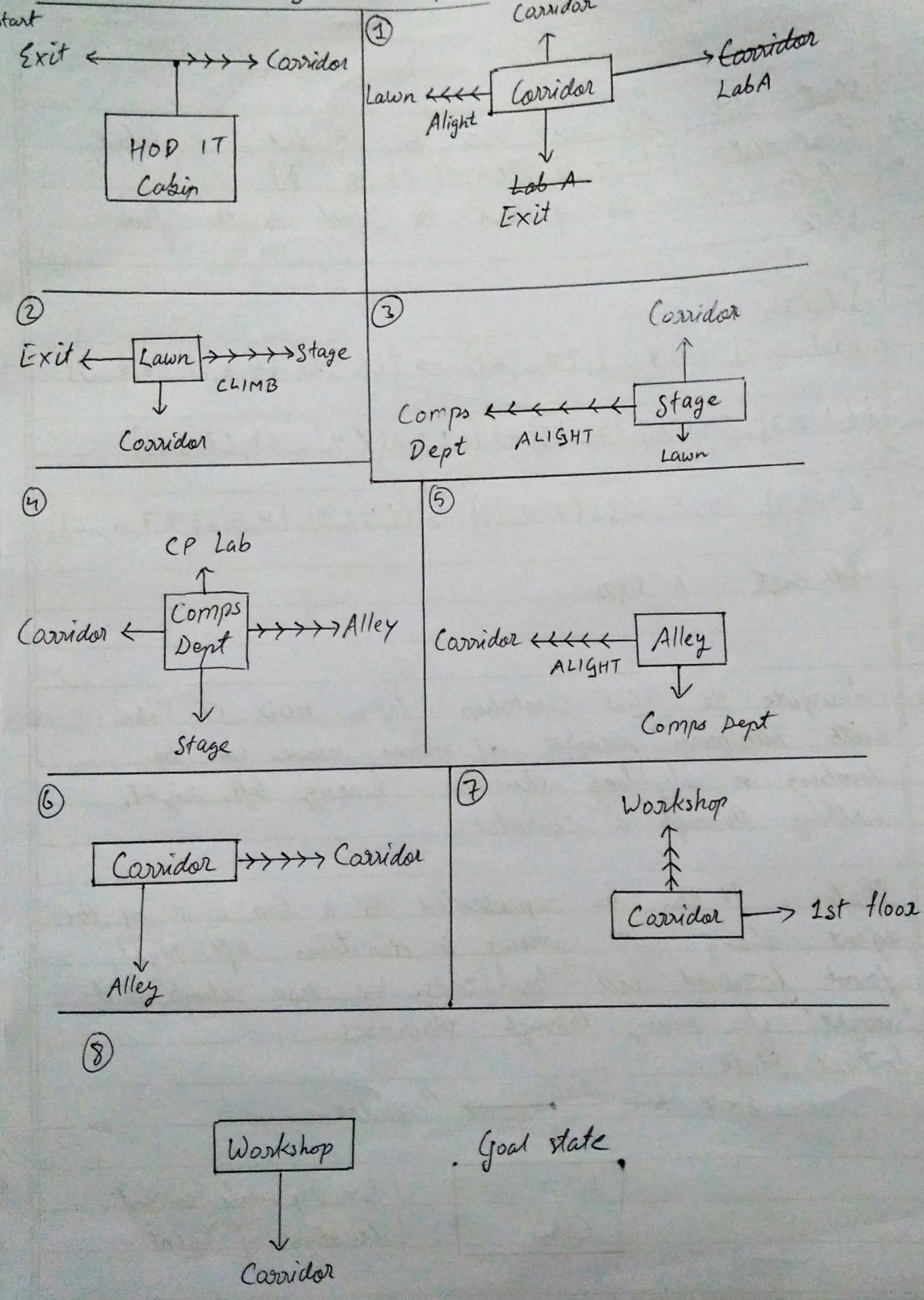
(ii) Navigate to KGCE Workshop from HOD IT Cabin with minimum number of moves, moves can be climbing or alighting staircase, turning left, right, walking through a corridor.

- States: It can be represented as a top view of the agent along with arrows in directions left, right, front forward and backwards. We use 'climb' and 'alight' for moving through staircases.

1 Initial State:



HOD IT Cabin → KGCE Workshop (Solution)

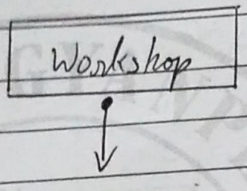


Goal state.

2 Actions : The agent moves in left, right, forward and backward directions along with alighting and climbing the stairs (if any).

3 Successor function : If we apply 'right' operation to the start state, the agent enters the corridor - the first step towards goal state.

4 Goal test



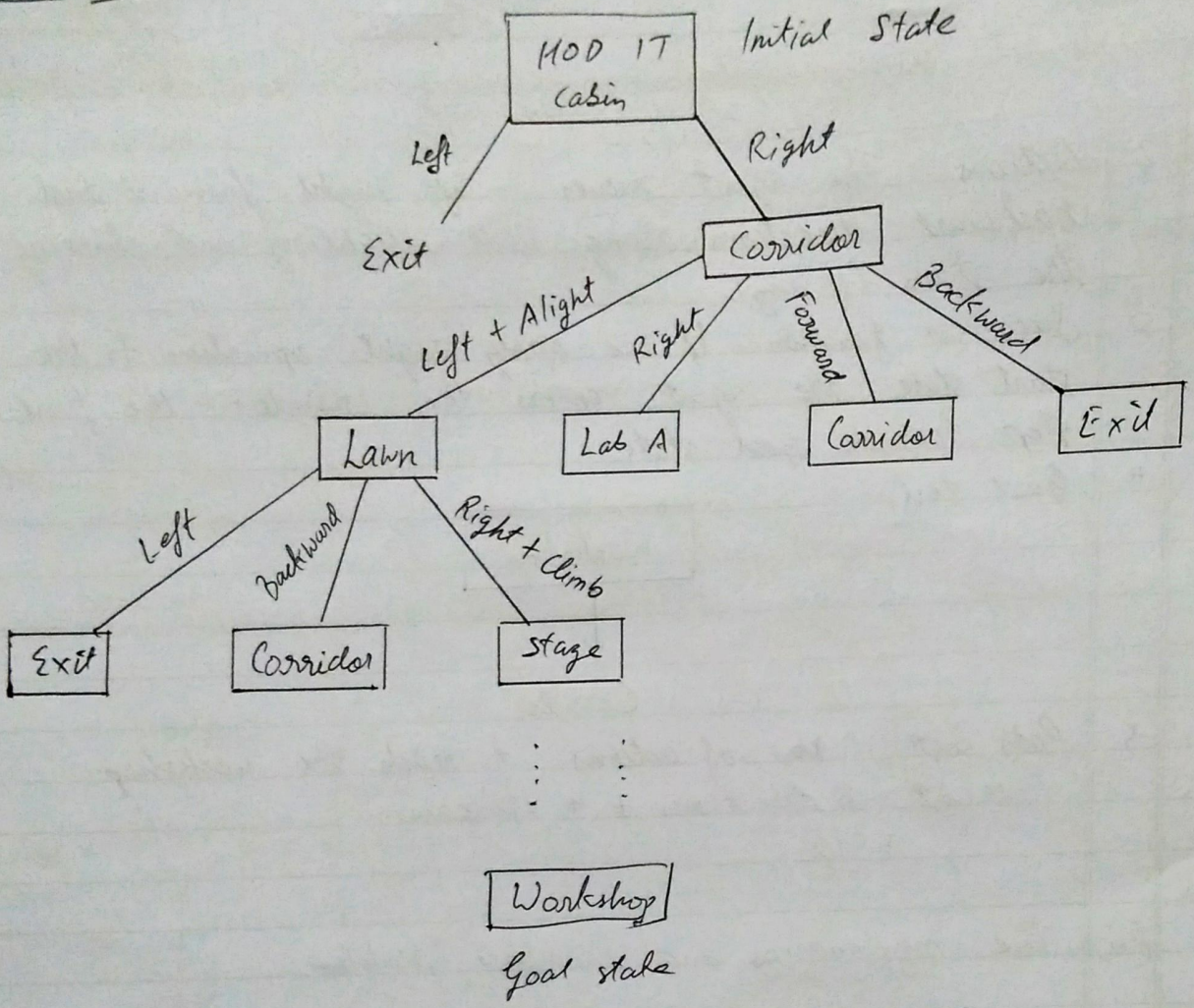
5 Path cost : No. of actions to reach the workshop.
Path cost = 8 directions + 4 staircases
= 12

(iii) The missionaries and cannibals problem.

States : State can be data structure having Triplet (i, j, k) representing the number of missionaries, cannibals and canoes on the left bank of the river respectively.

1. Initial state : (3, 3, 1) as all missionaries, cannibals and canoes are on the left bank
2. Actions : Take x number of missionaries & y number of cannibals.
3. Successor function : If we take one missionary, one cannibal the other side of the river will have two missionaries and cannibals left.
4. Goal Test : (0, 0, 0)
5. Path cost : Number of crossings to attain goal state

State Space



Workshop

Goal state

Solution:

(3,3,1) → (2,2,0) → (3,2,1) → (3,0,0) → (3,1,1) → (1,1,0)
→ (2,2,1) → (0,2,0) → (0,3,1) → (0,1,0) → (0,2,1) → (0,0,0)

Cost = 11 crossings