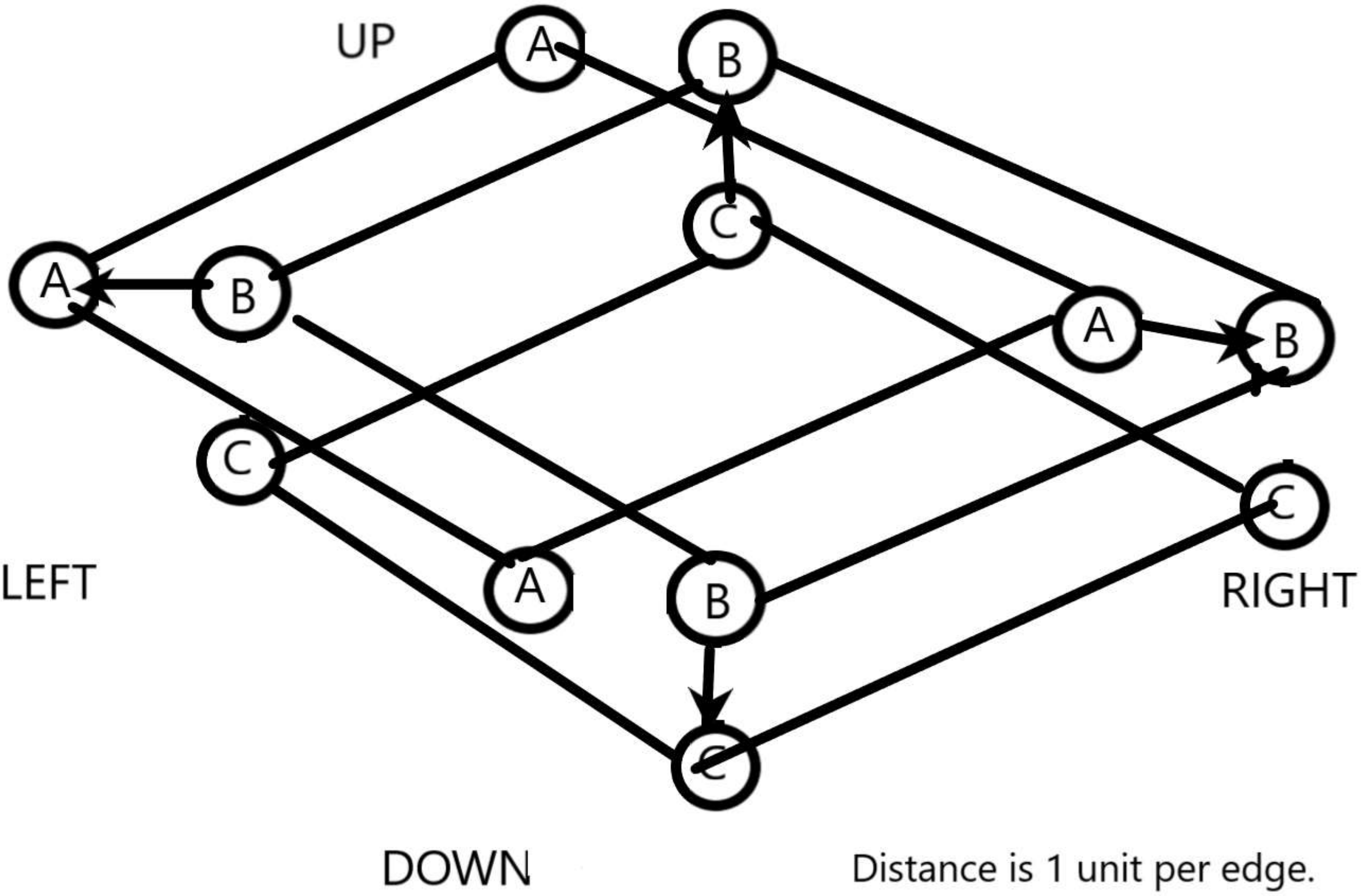
Practice Problems – Set 1

(Topics: Path-finding, State Machine, Decision Tree Behaviour Tree)

1

a



b

Here, we use `_u` to represent up, `_d` for down, `_l` for left and `_r` for right. For example: `A_u` means NPC is at A and faces up (north).

1. 2:A_u
2. 3:A_u->A_r
3. 3:A_r->B_r, 4:A_r->A_d
4. 4:B_r->B_d, 4:A_r->A_d
5. 4:B_d->C_d, 5:A_d->A_l

So the path is: A_u->A_r->B_r->B_d->C_d

A* algorithm needs 5 iterations.

c

i) Yes, it is. For Manhattan distance does not consider the cost of turning. It makes Manhattan distance always smaller than real distance.

ii) Yes. For combined distance+heuristic from the initial vertex never decreases along any path. h(A)≤ cost(A,B)+h(B) always holds for h(A)=Md(A)<=Md(A,B)+Md(B)<=cost(a,b)+MD(b)

2

a)

Using shortest path as Heuristic function.

-	C1	C2	C3	C4
C1	0	12	28	15
C2	12	0	16	27
C3	28	16	0	3
C4	15	27	3	0

b)

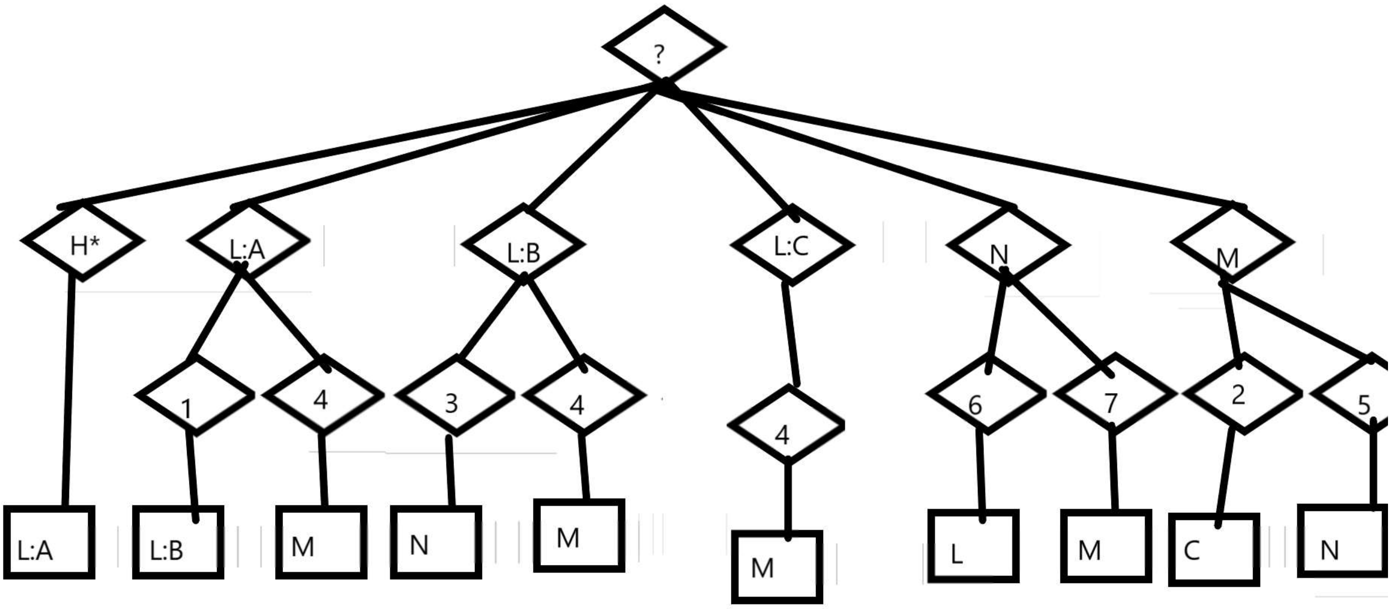
1. 28:S
2. 36:S->A, 37:S->B
3. 37:S->B, 39:A->D, 40:A->B
4. 39:A->D, 39:B->M, 40:B->C
5. 39:B->M, 40:B->C, 41:D->C
6. 40:B->C, 41:D->C, 41:M->O, 42:M->K, 51:M->N
7. 30:C->Q, 40:C->K, 41:M->O, 42:M->K, 51:M->N
8. 40:C->K, 42:M->K, 41:M->O, 51:M->N, 41:Q->E, 35:Q->F
9. 40:C->K, 42:M->K, 41:M->O, 51:M->N, 41:Q->E, 38:F->G
10. 40:C->K, 42:M->K, 41:M->O, 51:M->N, 41:Q->E, 40:G->E
11. 40:C->K, 42:M->K, 41:M->O, 51:M->N, 40:E->J
12. 49:K->L, 42:J->H, 41:M->O, 51:M->N
13. 49:K->L, 50:O->**I**, 61:O->N, 51:M->N, 42:J->H
14. 49:K->L, 50:O->**I**, 61:O->N, 51:M->N, 49:H->**I**
15. 49: **I**

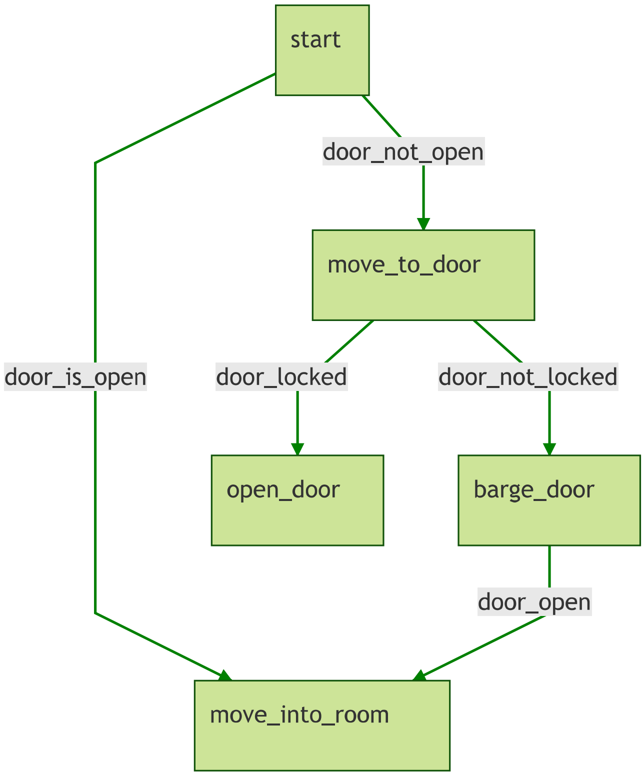
3

a)

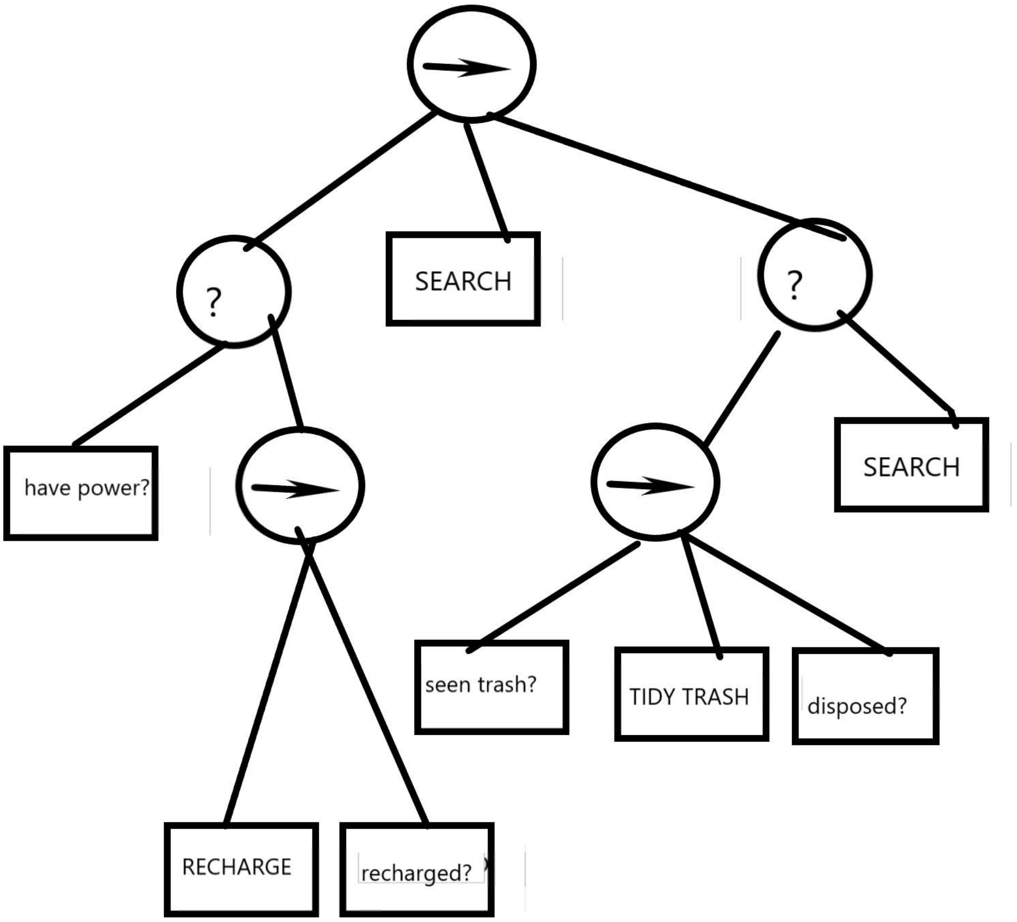
Target State: L:C
Action: M_exit + 2_action + L_entry + C_entry

b)


?
H*
L:A
L:B
L:C
N
M
1
4
3
4
4
6
7
2
5
L:A
L:B
M
N
M
M
L
M
C
N


start
door_not_open
door_is_open
move_to_door
door_locked
door_not_locked
open_door
barge_door
door_open
move_into_room

5


?
SEARCH
?
have power?
SEARCH
seen trash?
TIDY TRASH
disposed?
RECHARGE
recharged?