PAGE NO :
DATE :

Op. : Enter 3x3 puzzle board

Row 1 : 1 2 3

Row 2 : 4 0 5

Row 3 : 7 8 6

1	2	3		1	2	3		1	2	3
4	0	5	→	4	5	0	→	4	5	6
7	8	6		7	8	6		7	8	0

Manhattan Dist:

→ Tile 1 :

current (0, 0)

goal (0,0)

Distance = (0-0) + (1-1) = 0

→ Tile 2 :

current (0, 1)

goal (0, 1)

Distance = (0-0) + (1-1) = 0

→ Tile 3 :

current (0, 2)

goal (0, 2)

Distance = (0-0) + (2-2) = 0

Tile 4 :

current (1, 0)

goal (1, 0)

Distance (1-1) + (0-0) = 0.

Tile 5 : current (1, 1)

goal (2, 2)

Distance = |1-2| + |1-2| = 2;

18/11 25/11

Title 6:

current (1,2)
goal (1,1)
Distance = (1-1) + (1-2) = 1

→ Title 7:

current (2,0)
goal (2,0)
Distance = 0.

→ Title 8

current (2,1)
goal (2,1)
distance = 0

Title 9

current (2,2)
goal (1,2)
distance = (1-2) + (2-2) ⇒ 1

PAGE NO :
DATE :

18| 25|10|24

LAB - 3

A* algorithm

function A* search (problem) returns a solution or failure.

node ← a node n with n state:
problem. initial state

frontier ← a priority queue ordered by ascending g(n) only element

loop do
if empty > (frontier) then return failure
n ← pop (frontier)
if problem. goalTest (n. state) then return solution(n)
for each action a in problem,
actions (n. state) do
n' ← childNode (problem, n, a)
insert (n', g(n') + h(n'), frontier)