EXPERIMENT – 10 A\* Algorithm (grid with Manhattan)

AIM:

Path find on a grid from start to goal.

CODE:

# astar\_grid.py

import heapq

def astar(grid,start,goal):

rows=len(grid); cols=len(grid[0])

def h(a,b): return abs(a[0]-b[0])+abs(a[1]-b[1])

open\_heap=[(h(start,goal),0,start,None)]

came={}; g={start:0}

while open\_heap:

f, gc, cur, parent = heapq.heappop(open\_heap)

if cur in came: continue

came[cur]=parent

if cur==goal:

path=[]; p=cur

while p: path.append(p); p=came[p]

return list(reversed(path))

for dr,dc in [(1,0),(-1,0),(0,1),(0,-1)]:

nr, nc = cur[0]+dr, cur[1]+dc

if 0<=nr<rows and 0<=nc<cols and grid[nr][nc]==0:

ng=gc+1

nb=(ng+h((nr,nc),goal),ng,(nr,nc),cur)

if (nr,nc) not in g or ng<g[(nr,nc)]:

g[(nr,nc)]=ng; heapq.heappush(open\_heap,nb)

return None

if \_\_name\_\_=='\_\_main\_\_':

grid=[[0,0,0,0],[1,1,0,1],[0,0,0,0],[0,1,1,0]]

print(astar(grid,(0,0),(3,3)))

OUTPUT:

