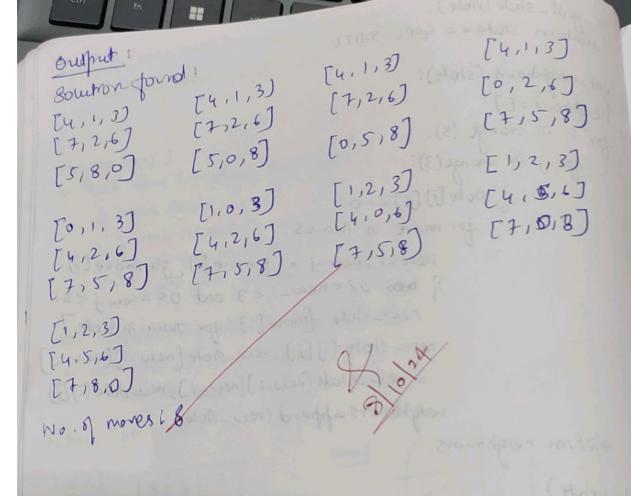


Algorithm (using montation dist) DPS_manbothon (short, goal): neighbours of) I more (new) of more (new) I new is not in visited. dist, = manhatron (next, goal) (House of odd (new, distones) to neighbours the Bort neighbourn by diretone (lougest fixt) for each Crew, dist in neighbours: add to stock sehm no solution lumal: del (store to a odd (seef stole) unistate == good_state) section moves Code: from collections import deque CLOPAL_STATE = [[1,2,3], [4,5,6], [78,0] roves = [(-1,0), (1,0), (0,-1), (0,1) montotron - distance (state): obstance = 0 for i in monge (3): for j in songe (3): 1 store (1) [1) =0 go al-i, goal-j = dirmod (state [i][j]-1,3)
distance + = abs (i-goal-i) + abs (j-goal-j) gream distance

```
del 18-goal_state (state):
        oration state = GONL_STATE
   det get-neighbour (state):
       neighbous = []
       for: m storge (3):
          for in orange (3):
           7 shote [i] (i) == 0:
          for move in MOVES:
                         new-i, new-j=i+more[],j+more[]
                        of new or new-i < 3 and o < = rew j < 3:
                          new_state =[010w[:7 for now in state]
                         new-state [][i], new-state [new-i][new-j]
                          = new state [new:][new-i), new state [][]
                        neighbours append (new - Stake)
         grehm neighbours.
 de des (state):
     quem = dequem ([(snote,[snoke])])
     visited = set ()
     while queue:
        current-snote, path = queue.popleft(14
        16 18 goal_state (wrent-state):
        of tuple (mop Chiple, current, shoke)) in visited:
       visited odd (hiple (rop (hiple ; ument -shok)))
  July None
 mmal_shate =[[4,1,3],
                    [7,2,6],
                    [5,8,0]]
parm = des (missal-state)
      print ("golution found)
Il poth:
      for shate in path.
         for grow in state:

point (now)
```



content work hope a chere policely 19

story in plats